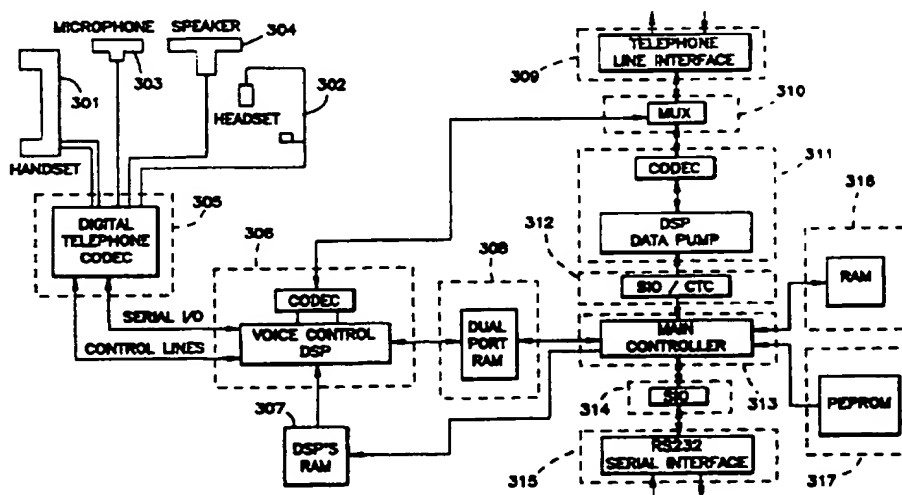




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : H04M 11/06, G10L 9/14	A1	(11) International Publication Number: WO 96/17465 (43) International Publication Date: 6 June 1996 (06.06.96)
(21) International Application Number: PCT/US95/14826 (22) International Filing Date: 13 November 1995 (13.11.95) (30) Priority Data: 08/346,421 29 November 1994 (29.11.94) US (71) Applicant: MULTI-TECH SYSTEMS, INC. [US/US]; 2205 Woodale Drive, Mounds View, MN 55112 (US). (72) Inventors: SHARMA, Raghu, N.; 16 Evergreen Road, North Oaks, MN 55127 (US). DAVIS, Jeffrey, P.; 3244 146th Avenue, Ham Lake, MN 55304 (US). GUNN, Timothy, D.; 8120 Red Oak Court, Mounds View, MN 55432 (US). LI, Ping; 694 4th Avenue, New Brighton, MN 55112 (US). MAITRA, Sidhartha; 13750 Surrey Lane, Saratoga, CA 95070 (US). THANAWALA, Ashish, A.; Unit 25, 20812 Fourth Street, Saratoga, CA 95070 (US). (74) Agent: VIKSNINS, Ann, S.; Schwegman, Lundberg & Woessner, P.O. Box 2938, Minneapolis, MN 55402 (US).		(81) Designated States: CA, JP, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: DYNAMIC SELECTION OF VOICE COMPRESSION RATE IN A VOICE DATA MODEM



(57) Abstract

The present invention includes software and hardware components to enable digital data communication over standard telephone lines. The present invention converts analog voice signals to digital data, compresses that data and places the compressed speech data into packets for transfer over the telephone lines to a remote site. A voice control digital signal processor (DSP) operates to use one of a plurality of speech compression algorithms which produce a scaleable amount of compression. The rate of compression is inversely proportional to the quality of the speech the compression algorithm is able to reproduce. The higher the compression, the lower the reproduction quality. The selection of the rate of compression is dependent on such factors as the speed or data bandwidth on the communications connection between the two sites, the data demand between the sites and amount of silence detected in the speech signal. The voice compression rate is dynamically changed as the aforementioned factors change. A negotiation handshake protocol is described which enables the two sites to negotiate the compression rate based on such factors.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	GB	United Kingdom	MR	Mauritania
AU	Australia	GE	Georgia	MW	Malawi
BB	Barbados	GN	Guinea	NE	Niger
BE	Belgium	GR	Greece	NL	Netherlands
BF	Burkina Faso	HU	Hungary	NO	Norway
BG	Bulgaria	IE	Ireland	NZ	New Zealand
BJ	Benin	IT	Italy	PL	Poland
BR	Brazil	JP	Japan	PT	Portugal
BY	Belarus	KE	Kenya	RO	Romania
CA	Canada	KG	Kyrgyzstan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo	KR	Republic of Korea	SE	Sweden
CH	Switzerland	KZ	Kazakhstan	SI	Slovenia
CI	Côte d'Ivoire	LI	Liechtenstein	SK	Slovakia
CM	Cameroon	LK	Sri Lanka	SN	Senegal
CN	China	LU	Luxembourg	TD	Chad
CS	Czechoslovakia	LV	Latvia	TG	Togo
CZ	Czech Republic	MC	Monaco	TJ	Tajikistan
DE	Germany	MD	Republic of Moldova	TT	Trinidad and Tobago
DK	Denmark	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	US	United States of America
FI	Finland	MN	Mongolia	UZ	Uzbekistan
FR	France			VN	Viet Nam
GA	Gabon				

DYNAMIC SELECTION OF VOICE COMPRESSION
RATE IN A VOICE DATA MODEM

5

Field of the Invention

The present invention relates to communications systems and in particular to modem communications having simultaneous digitized voice and data capabilities.

10

Background of the Invention

Simultaneous voice and modem data transmitted over the same communications link between two sites has been accomplished in several ways. The most common communications link used between two sites is the telephone line. The most common data handling equipment to communicate over a communications link is the computer modem which modulates digital data onto a carrier for transmission in the voice band of the telephone line. A wide variety of modulation standards have been promulgated by such international groups as the CCITT for communication in the voice band. The data bandwidth for such modulation standards is typically fixed and the throughput rate of data is also assumed to be fixed.

In some modulations standards, there are provisions for changing the modulation data rate based on the quality of the communications link. For example, in a noisy telephone line, a 9600 baud modulation rate may have such a high bit error rate that the modulation must be changed to a 2400 baud connection. This is done in a handshake communication protocol between the two sites when the communications link simply cannot support the higher rate.

There is a need in the art, however, for an efficient and cost effective way of maximizing bandwidth over the communications link between two sites to enable the simultaneous transmission of voice and data. There is a need, therefore, to negotiate the data bandwidth between the sites, negotiate the compression rate for the voice compression algorithms used to

compress the voice and there is a need to allocate and reallocate the ratio of compress voice to digital data transmitted over the communications link.

Summary of the Invention

The present invention solves the aforementioned problems and
5 shortcoming of the existing art and solves other problems not listed above which will become apparent to those skilled in the art upon reading and understanding the present specification and claims. The present invention describes a voice over data modem which allows the operator to simultaneously transmit voice and data communication to a remote site. This
10 voice over data function dynamically allocates data bandwidth over the telephone line depending on the demands of the voice grade digitized signal and the modulation speed of the communication link between the two sites.

The present invention includes software and hardware components to enable digital data communication over standard telephone
15 lines. The present invention converts analog voice signals to digital data, compresses that data and places the compressed speech data into packets for transfer over the telephone lines to a remote site. A voice control digital signal processor (DSP) operates to use one of a plurality of speech compression algorithms which produce a scaleable amount of compression.
20 The rate of compression is inversely proportional to the quality of the speech the compression algorithm is able to reproduce. The higher the compression, the lower the reproduction quality. The selection of the rate of compression is dependant on such factors as the speed or data bandwidth on the communications connection between the two sites, the data demand between
25 the sites and amount of silence detected in the speech signal. The voice compression rate is dynamically changed as the aforementioned factors change. A negotiation handshake protocol is described which enables the two sites to negotiate the compression rate based on such factors.

Description of the Drawings

30 In the drawings, where like numerals describe like components throughout the several views,

Figure 1 shows the telecommunications environment within which the present may operate in several of the possible modes of communication;

Figure 2 is a block diagram of the hardware components of the present system;

Figure 3 is a detailed function flow diagram of the speech compression algorithm;

Figure 4 is a detailed function flow diagram of the speech decompression algorithm;

Figure 5 is a signal flow diagram of the speech compression algorithm;

Figure 6 is a signal flow diagram of the speech compression algorithm showing details of the code book synthesis; and

Figure 7 is a detailed function flow diagram of the voice/data multiplexing function.

Detailed Description of the Preferred Embodiments

In the following detailed description of the preferred embodiment, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific embodiments in which the inventions may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized and that structural changes may be made without departing from the spirit and scope of the present inventions. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present inventions is defined by the appended claims.

Figure 1 shows a typical arrangement for the use of the present system. Personal computer 10 is running the software components of the present system while the hardware components 20 include the data communication equipment and telephone headset. Hardware components 20 communicate over a standard telephone line 30 to one of a variety of remote sites. One of the remote sites may be equipped with the present system

including hardware components 20a and software components running on personal computer 10a. In one alternative use, the local hardware components 20 may be communicating over standard telephone line 30 to facsimile machine 60. In another alternative use, the present system may be
5 communicating over a standard telephone line 30 to another personal computer 80 through a remote modem 70. In another alternative use, the present system may be communicating over a standard telephone line 30 to a standard telephone 90. Those skilled in the art will readily recognize the wide variety of communication interconnections possible with the present system by
10 reading and understanding the following detailed description.

General Overview

The present inventions are embodied in a commercial product by the assignee, Multi-Tech Systems, Inc. The software component operating on a personal computer is sold under the commercial trademark of
15 MultiExpressPCSTTM personal communications software while the hardware component of the present system is sold under the commercial name of MultiModemPCSTTM, Intelligent Personal Communications System Modem. In the preferred embodiment, the software component runs under Microsoft[®] WindowsTM however those skilled in the art will readily recognize that the
20 present system is easily adaptable to run under any single or multi-user, single or multi-window operating system.

The present system is a multifunction communication system which includes hardware and software components. The system allows the user to connect to remote locations equipped with a similar system or with
25 modems, facsimile machines or standard telephones over a single analog telephone line. The software component of the present system includes a number of modules which are described in more detail below.

The telephone module allows the system to operate as a conventional or sophisticated telephone system. The system converts voice
30 into a digital signal so that it can be transmitted or stored with other digital data, like computer information. The telephone function supports PBX and Centrex features such a call waiting, call forwarding, caller ID and three-way

calling. This module also allows the user to mute, hold or record a conversation. The telephone module enables the handset, headset or hands-free speaker telephone operation of the hardware component. It includes on-screen push button dialing, speed-dial of stored numbers and digital recording of two-way conversations.

The voice mail portion of the present system allows this system to operate as a telephone answering machine by storing voice messages as digitized voice files along with a time/date voice stamp. The digitized voice files can be saved and sent to one or more destinations immediately or at a later time using a queue scheduler. The user can also listen to, forward or edit the voice messages which have been received with a powerful digital voice editing component of the present system. This module also creates queues for outgoing messages to be sent at preselected times and allows the users to create outgoing messages with the voice editor.

The fax manager portion of the present system is a queue for incoming and outgoing facsimile pages. In the preferred embodiment of the present system, this function is tied into the Windows "print" command once the present system has been installed. This feature allows the user to create faxes from any Windows[®]-based document that uses the "print" command. The fax manager function of the present system allows the user to view queued faxes which are to be sent or which have been received. This module creates queues for outgoing faxes to be sent at preselected times and logs incoming faxes with time/date stamps.

The multi-media mail function of the present system is a utility which allows the user to compose documents that include text, graphics and voice messages using the message composer function of the present system, described more fully below. The multi-media mail utility of the present system allows the user to schedule messages for transmittal and queues up the messages that have been received so that can be viewed at a later time.

The show and tell function of the present system allows the user to establish a data over voice (DOV) communications session. When the user is transmitting data to a remote location similarly equipped, the user is

able to talk to the person over the telephone line while concurrently transferring the data. This voice over data function is accomplished in the hardware components of the present system. It digitizes the voice and transmits it in a dynamically changing allocation of voice data and digital data multiplexed in the same transmission. The allocation at a given moment is selected depending on the amount of voice digital information required to be transferred. Quiet voice intervals allocate greater space to the digital data transmission.

The terminal function of the present system allows the user to establish a data communications session with another computer which is equipped with a modem but which is not equipped with the present system. This feature of the present system is a WindowsTM-based data communications program that reduces the need for issuing "AT" commands by providing menu driven and "pop-up" window alternatives.

The address book function of the present system is a database that is accessible from all the other functions of the present system. This database is created by the user inputting destination addresses and telephone numbers for data communication, voice mail, facsimile transmission, modem communication and the like. The address book function of the present system may be utilized to broadcast communications to a wide variety of recipients. Multiple linked databases have separate address books for different groups and different destinations may be created by the users. The address book function includes a textual search capability which allows fast and efficient location of specific addresses as described more fully below.

Hardware Components

Figure 2 is a block diagram of the hardware components of the present system corresponding to reference number 20 of Figure 1. These components form the link between the user, the personal computer running the software component of the present system and the telephone line interface. As will be more fully described below, the interface to the hardware components of the present system is via a serial communications port connected to the personal computer. The interface protocol is well ordered

and defined such that other software systems or programs running on the personal computer may be designed and implemented which would be capable of controlling the hardware components shown in Figure 2 by using the control and communications protocol defined below.

5 In the preferred embodiment of the present system, three alternate telephone interfaces are available: the telephone handset 301, a telephone headset 302, and a hands-free microphone 303 and speaker 304. Regardless of the telephone interface, the three alternative interfaces connect to the digital telephone coder-decoder (CODEC) circuit 305.

10 The digital telephone CODEC circuit 305 interfaces with the voice control digital signal processor (DSP) circuit 306 which includes a voice control DSP and CODEC. This circuit does digital to analog (D/A) conversion, analog to digital (A/D) conversion, coding/decoding, gain control and is the interface between the voice control DSP circuit 306 and the
15 telephone interface. The CODEC of the voice control circuit 306 transfers digitized voice information in a compressed format to multiplexor circuit 310 to analog telephone line interface 309.

 The CODEC of the voice control circuit 306 is actually an integral component of a voice control digital signal processor integrated
20 circuit, as described more fully below. The voice control DSP of circuit 306 controls the digital telephone CODEC circuit 305, performs voice compression and echo cancellation.

 Multiplexor (MUX) circuit 310 selects between the voice control DSP circuit 306 and the data pump DSP circuit 311 for transmission
25 of information on the telephone line through telephone line interface circuit 309.

 The data pump circuit 311 also includes a digital signal processor (DSP) and a CODEC for communicating over the telephone line interface 309 through MUX circuit 310. The data pump DSP and CODEC of
30 circuit 311 performs functions such as modulation, demodulation and echo cancellation to communicate over the telephone line interface 309 using a

plurality of telecommunications standards including FAX and modem protocols.

The main controller circuit 313 controls the DSP data pump circuit 311 and the voice control DSP circuit 306 through serial input/output and clock timer control (SIO/CTC) circuits 312 and dual port RAM circuit 308 respectively. The main controller circuit 313 communicates with the voice control DSP 306 through dual port RAM circuit 308. In this fashion digital voice data can be read and written simultaneously to the memory portions of circuit 308 for high speed communication between the user (through interfaces 301, 302 or 303/304) and the personal computer connected to serial interface circuit 315 and the remote telephone connection connected through the telephone line attached to line interface circuit 309.

As described more fully below, the main controller circuit 313 includes, in the preferred embodiment, a microprocessor which controls the functions and operation of all of the hardware components shown in Figure 2. The main controller is connected to RAM circuit 316 and an programmable and electrically erasable read only memory (EEPROM) circuit 317. The EEPROM circuit 317 includes non-volatile memory in which the executable control programs for the voice control DSP circuits 306 and the main controller circuits 313 operate.

The RS232 serial interface circuit 315 communicates to the serial port of the personal computer which is running the software components of the present system. The RS232 serial interface circuit 315 is connected to a serial input/output circuit 314 with main controller circuit 313. SIO circuit 314 is in the preferred embodiment, a part of SIO/CTC circuit 312.

Functional Operation of the Hardware Components

Referring once again to Figure 2, the multiple and selectable functions described in conjunction with Figure 2 are all implemented in the hardware components of Figure 2. Each of these functions will be discussed in turn.

The telephone function 115 is implemented by the user either selecting a telephone number to be dialed from the address book 127 or

manually selecting the number through the telephone menu on the personal computer. The telephone number to be dialed is downloaded from the personal computer over the serial interface and received by main controller 313. Main controller 313 causes the data pump DSP circuit 311 to seize the telephone line and transmit the DTMF tones to dial a number. DSP 306 receives commands from the personal computer via main controller 313 to configure the digital telephone CODEC circuit 305 to enable either the handset 301 operation, the microphone 303 and speaker 304 operation or the headset 302 operation. A telephone connection is established through the telephone line interface circuit 309 and communication is enabled. The user's analog voice is transmitted in an analog fashion to the digital telephone CODEC 305 where it is digitized. The digitized voice patterns are passed to the voice control circuit 306 where echo cancellation is accomplished, the digital voice signals are reconstructed into analog signals and passed through multiplexor circuit 310 to the telephone line interface circuit 309 for analog transmission over the telephone line. The incoming analog voice from the telephone connection through telephone connection circuit 309 is passed to the integral CODEC of the voice control circuit 306 where it is digitized. The digitized incoming voice is then passed to digital telephone CODEC circuit 305 where it is reconverted to an analog signal for transmission to the selected telephone interface (either the handset 301, the microphone/speaker 303/304 or the headset 302). Voice Control DSP circuit 306 is programmed to perform echo cancellation to avoid feedback and echoes between transmitted and received signals, as is more fully described below.

In the voice mail function mode of the present system, voice messages may be stored for later transmission or the present system may operate as an answering machine receiving incoming messages. For storing digitized voice, the telephone interface is used to send the analog speech patterns to the digital telephone CODEC circuit 305. Circuit 305 digitizes the voice patterns and passes them to voice control circuit 306 where the digitized voice patterns are digitally compressed. The digitized and compressed voice patterns are passed through dual port ram circuit 308 to the main controller

circuit 313 where they are transferred through the serial interface to the personal computer using a packet protocol defined below. The voice patterns are then stored on the disk of the personal computer for later use in multimedia mail, for voice mail, as a pre-recorded answering machine message or
5 for later predetermined transmission to other sites.

For the present system to operate as an answering machine, the hardware components of Figure 2 are placed in answer mode. An incoming telephone ring is detected through the telephone line interface circuit 309 and the main controller circuit 313 is alerted which passes the information off to
10 the personal computer through the RS232 serial interface circuit 315. The telephone line interface circuit 309 seizes the telephone line to make the telephone connection. A pre-recorded message may be sent by the personal computer as compressed and digitized speech through the RS232 interface to the main controller circuit 313. The compressed and digitized speech from
15 the personal computer is passed from main controller circuit 313 through dual port ram circuit 308 to the voice control DSP circuit 306 where it is uncompressed and converted to analog voice patterns. These analog voice patterns are passed through multiplexor circuit 310 to the telephone line interface 309 for transmission to the caller. Such a message may invite the
20 caller to leave a voice message at the sound of a tone. The incoming voice messages are received through telephone line interface 309 and passed to voice control circuit 306. The analog voice patterns are digitized by the integral CODEC of voice control circuit 306 and the digitized voice patterns are compressed by the voice control DSP of the voice control circuit 306.
25 The digitized and compressed speech patterns are passed through dual port ram circuit 308 to the main controller circuit 313 where they are transferred using packet protocol described below through the RS232 serial interface 315 to the personal computer for storage and later retrieval. In this fashion the hardware components of Figure 2 operate as a transmit and receive voice mail
30 system for implementing the voice mail function 117 of the present system.

The hardware components of Figure 2 may also operate to facilitate the fax manager function 119 of Figure 2. In fax receive mode, an

incoming telephone call will be detected by a ring detect circuit of the telephone line interface 309 which will alert the main controller circuit 313 to the incoming call. Main controller circuit 313 will cause line interface circuit 309 to seize the telephone line to receive the call. Main controller circuit 313 will also concurrently alert the operating programs on the personal computer through the RS232 interface using the packet protocol described below. Once the telephone line interface seizes the telephone line, a fax carrier tone is transmitted and a return tone and handshake is received from the telephone line and detected by the data pump circuit 311. The reciprocal transmit and receipt of the fax tones indicates the imminent receipt of a facsimile transmission and the main controller circuit 313 configures the hardware components of Figure 2 for the receipt of that information. The necessary handshaking with the remote facsimile machine is accomplished through the data pump 311 under control of the main controller circuit 313. The incoming data packets of digital facsimile data are received over the telephone line interface and passed through data pump circuit 311 to main controller circuit 313 which forwards the information on a packet basis (using the packet protocol described more fully below) through the serial interface circuit 315 to the personal computer for storage on disk. Those skilled in the art will readily recognize that the FAX data could be transferred from the telephone line to the personal computer using the same path as the packet transfer except using the normal AT stream mode. Thus the incoming facsimile is automatically received and stored on the personal computer through the hardware components of Figure 2.

A facsimile transmission is also facilitated by the hardware components of Figure 2. The transmission of a facsimile may be immediate or queued for later transmission at a predetermined or preselected time. Control packet information to configure the hardware components to send a facsimile are sent over the RS232 serial interface between the personal computer and the hardware components of Figure 2 and are received by main controller circuit 313. The data pump circuit 311 then dials the recipient's telephone number using DTMF tones or pulse dialing over the telephone line

interface circuit 309. Once an appropriate connection is established with the remote facsimile machine, standard facsimile handshaking is accomplished by the data pump circuit 311. Once the facsimile connection is established, the digital facsimile picture information is received through the data packet
5 protocol transfer over serial line interface circuit 315, passed through main controller circuit 313 and data pump circuit 311 onto the telephone line through telephone line interface circuit 309 for receipt by the remote facsimile machine.

The operation of the multi-media mail function 121 of Figure 2
10 is also facilitated by the hardware components of Figure 2. A multimedia transmission consists of a combination of picture information, digital data and digitized voice information. For example, the type of multimedia information transferred to a remote site using the hardware components of Figure 2 could be the multimedia format of the MicroSoft® Multimedia Wave® format with
15 the aid of an Intelligent Serial Interface (ISI) card added to the personal computer. The multimedia may also be the type of multimedia information assembled by the software component of the present system which is described more fully below.

The multimedia package of information including text, graphics
20 and voice messages (collectively called the multimedia document) may be transmitted or received through the hardware components shown in Figure 2. For example, the transmission of a multimedia document through the hardware components of Figure 2 is accomplished by transferring the multimedia digital information using the packet protocol described below over
25 the RS232 serial interface between the personal computer and the serial line interface circuit 315. The packets are then transferred through main controller circuit 313 through the data pump circuit 311 on to the telephone line for receipt at a remote site through telephone line interface circuit 309. In a similar fashion, the multimedia documents received over the telephone line
30 from the remote site are received at the telephone line interface circuit 309, passed through the data pump circuit 311 for receipt and forwarding by the main controller circuit 313 over the serial line interface circuit 315.

The show and tell function 123 of the present system allows the user to establish a data over voice communication session. In this mode of operation, full duplex data transmission may be accomplished simultaneously with the voice communication between both sites. This mode of operation
5 assumes a like- configured remote site. The hardware components of the present system also include a means for sending voice/data over cellular links. The protocol used for transmitting multiplexed voice and data include a supervisory packet described more fully below to keep the link established through the cellular link. This supervisory packet is an acknowledgement that
10 the link is still up. The supervisory packet may also contain link information to be used for adjusting various link parameters when needed. This supervisory packet is sent every second when data is not being sent and if the packet is not acknowledged after a specified number of attempts, the protocol would then give an indication that the cellular link is down and then allow the
15 modem to take action. The action could be for example; change speeds, retrain, or hang up. The use of supervisory packets is a novel method of maintaining inherently intermittent cellular links when transmitting multiplexed voice and data.

The voice portion of the voice over data transmission of the
20 show and tell function is accomplished by receiving the user's voice through the telephone interface 301, 302 or 303 and the voice information is digitized by the digital telephone circuit 305. The digitized voice information is passed to the voice control circuit 306 where the digitized voice information is compressed using a voice compression algorithm described more fully below.
25 The digitized and compressed voice information is passed through dual port RAM circuit 308 to the main controller circuit 313. During quiet periods of the speech, a quiet flag is passed from voice control circuit 306 to the main controller 313 through a packet transfer protocol described below by a dual port RAM circuit 308.

30 Simultaneous with the digitizing compression and packetizing of the voice information is the receipt of the packetized digital information from the personal computer over interface line circuit 315 by main controller

circuit 313. Main controller circuit 313 in the show and tell function of the present system must efficiently and effectively combine the digitized voice information with the digital information for transmission over the telephone line via telephone line interface circuit 309. As described above and as
5 described more fully below, main controller circuit 313 dynamically changes the amount of voice information and digital information transmitted at any given period of time depending upon the quiet times during the voice transmissions. For example, during a quiet moment where there is no speech information being transmitted, main controller circuit 313 ensures that a
10 higher volume of digital data information be transmitted over the telephone line interface in lieu of digitized voice information.

Also, as described more fully below, the packets of digital data transmitted over the telephone line interface with the transmission packet protocol described below, requires 100 percent accuracy in the transmission of
15 the digital data, but a lesser standard of accuracy for the transmission and receipt of the digitized voice information. Since digital information must be transmitted with 100 percent accuracy, a corrupted packet of digital information received at the remote site must be re-transmitted. A retransmission signal is communicated back to the local site and the packet of
20 digital information which was corrupted during transmission is retransmitted. If the packet transmitted contained voice data, however, the remote site uses the packets whether they were corrupted or not as long as the packet header was intact. If the header is corrupted, the packet is discarded. Thus, the voice information may be corrupted without requesting retransmission since it
25 is understood that the voice information must be transmitted on a real time basis and the corruption of any digital information of the voice signal is not critical. In contrast to this the transmission of digital data is critical and retransmission of corrupted data packets is requested by the remote site.

The transmission of the digital data follows the CCITT V.42
30 standard, as is well known in the industry and as described in the CCITT Blue Book, volume VIII entitled Data Communication over the Telephone Network, 1989. The voice data packet information also follows the CCITT V.42

standard, but uses a different header format so the receiving site recognizes the difference between a data packet and a voice packet. The voice packet is distinguished from a data packet by using undefined bits in the header (80 hex) of the V.42 standard. The packet protocol for voice over data

5 transmission during the show and tell function of the present system is described more fully below.

Since the voice over data communication with the remote site is full-duplex, incoming data packets and incoming voice packets are received by the hardware components of Figure 2. The incoming data packets and

10 voice packets are received through the telephone line interface circuit 309 and passed to the main controller circuit 313 via data pump DSP circuit 311. The incoming data packets are passed by the main controller circuit 313 to the serial interface circuit 315 to be passed to the personal computer. The incoming voice packets are passed by the main controller circuit 313 to the

15 dual port RAM circuit 308 for receipt by the voice control DSP circuit 306. The voice packets are decoded and the compressed digital information therein is uncompressed by the voice control DSP of circuit 306. The uncompressed digital voice information is passed to digital telephone CODEC circuit 305 where it is reconverted to an analog signal and retransmitted through the

20 telephone line interface circuits. In this fashion full-duplex voice and data transmission and reception is accomplished through the hardware components of Figure 2 during the show and tell functional operation of the present system.

Terminal operation 125 of the present system is also supported

25 by the hardware components of Figure 2. Terminal operation means that the local personal computer simply operates as a "dumb" terminal including file transfer capabilities. Thus no local processing takes place other than the handshaking protocol required for the operation of a dumb terminal. In terminal mode operation, the remote site is assumed to be a modem connected

30 to a personal computer but the remote site is not necessarily a site which is configured according to the present system. In terminal mode of operation, the command and data information from personal computer is transferred over

the RS232 serial interface circuit 315, forwarded by main controller circuit 313 to the data pump circuit 311 where the data is placed on the telephone line via telephone line interface circuit 309.

In a reciprocal fashion, data is received from the telephone line over telephone line interface circuit 309 and simply forwarded by the data pump circuit 311, the main controller circuit 313 over the serial line interface circuit 315 to the personal computer.

As described above, and more fully below, the address book function of the present system is primarily a support function for providing telephone numbers and addresses for the other various functions of the present system.

Packet Protocol Between the PC and the Hardware Component

A special packet protocol is used for communication between the hardware components 20 and the personal computer (PC) 10. The protocol is used for transferring different types of information between the two devices such as the transfer of DATA, VOICE, and QUALIFIED information. The protocol also uses the BREAK as defined in CCITT X.28 as a means to maintain protocol synchronization. A description of this BREAK sequence is also described in the Statutory Invention Registration entitled "ESCAPE METHODS FOR MODEM COMMUNICATIONS", to Timothy D. Gunn filed January 8, 1993.

The protocol has two modes of operation. One mode is packet mode and the other is stream mode. The protocol allows mixing of different types of information into the data stream without having to physically switch modes of operation. The hardware component 20 will identify the packet received from the computer 10 and perform the appropriate action according to the specifications of the protocol. If it is a data packet, then the controller 313 of hardware component 20 would send it to the data pump circuit 311. If the packet is a voice packet, then the controller 313 of hardware component 20 would distribute that information to the Voice DSP 306. This packet transfer mechanism also works in the reverse, where the controller 313 of

hardware component 20 would give different information to the computer 10 without having to switch into different modes. The packet protocol also allows commands to be sent to either the main controller 313 directly or to the Voice DSP 306 for controlling different options without having to enter a command state.

Packet mode is made up of 8 bit asynchronous data and is identified by a beginning synchronization character (01 hex) followed by an ID/LI character and then followed by the information to be sent. In addition to the ID/LI character codes defined below, those skilled in the art will readily recognize that other ID/LI character codes could be defined to allow for additional types of packets such as video data, or alternate voice compression algorithm packets such as Codebook Excited Linear Predictive Coding (CELP) algorithm, GSM, RPE, VSELP, etc.

Stream mode is used when large amounts of one type of packet (VOICE, DATA, or QUALIFIED) is being sent. The transmitter tells the receiver to enter stream mode by a unique command. Thereafter, the transmitter tells the receiver to terminate stream mode by using the BREAK command followed by an "AT" type command. The command used to terminate the stream mode can be a command to enter another type of stream mode or it can be a command to enter back into packet mode.

Currently there are 3 types of packets used: DATA, VOICE, and QUALIFIED. Table 1 shows the common packet parameters used for all three packet types. Table 2 shows the three basic types of packets with the sub-types listed.

25

TABLE 1: Packet Parameters

1. Asynchronous transfer
2. 8 bits, no parity
3. Maximum packet length of 128 bytes
 - Identifier byte = 1
 - InFormation = 127

30

4. SPEED

- variable from 9600 to 57600
- default to 19200

5

TABLE 2: Packet Types

1. Data
2. Voice
3. Qualified:
 - 10 a. COMMAND
 - b. RESPONSE
 - c. STATUS
 - d. FLOW CONTROL
 - e. BREAK
 - 15 f. ACK
 - g. NAK
 - h. STREAM

A Data Packet is shown in Table 1 and is used for normal data transfer between the controller 313 of hardware component 20 and the computer 10 for such things as text, file transfers, binary data and any other type of information presently being sent through modems. All packet transfers begin with a synch character 01 hex (synchronization byte). The Data Packet begins with an ID byte which specifies the packet type and packet length. Table 3 describes the Data Packet byte structure and Table 4 describes the bit structure of the ID byte of the Data Packet. Table 5 is an example of a Data Packet with a byte length of 6. The value of the LI field is the actual length of the data field to follow, not counting the ID byte.

TABLE 3: Data Packet Byte Structure

byte 1 = 01h (sync byte)
byte 2 = ID/LI (ID byte/length indicator)
bytes 3-127 = data (depending on LI)

01	ID					
SYNC	LI	data	data	data	data	data

TABLE 4: ID Byte of Data Packet

Bit 7 identifies the type of packet
Bits 6 - 0 contain the LI or length indicator portion of the ID byte

7	6	5	4	3	2	1	0
0	LI (Length Indicator) = 1 to 127						

TABLE 5: Data Packet Example

LI (length indicator) = 6

01	06						
SYNC	ID	data	data	data	data	data	data

The Voice Packet is used to transfer compressed VOICE messages between the controller 313 of hardware component 20 and the computer 10. The Voice Packet is similar to the Data Packet except for its length which is, in the preferred embodiment, currently fixed at 23 bytes of data. Once again, all packets begin with a synchronization character chosen in the preferred embodiment to be 01 hex (01H). The ID byte of the Voice Packet is completely a zero byte: all bits are set to zero. Table 6 shows the ID byte of the Voice Packet and Table 7 shows the Voice Packet byte structure.

TABLE 6: ID Byte of Voice Packet

	7	6	5	4	3	2	1	0
5	LI (Length Indicator) = 0							

TABLE 7: Voice Packet Byte Structure

	LI (length indicator) = 0							
	23 bytes of data							
	(depending upon compression algorithm used)							
20	01 SYNC	00 ID	data	data	data	data	data	

The Qualified Packet is used to transfer commands and other non-data/voice related information between the controller 313 of hardware component 20 and the computer 10. The various species or types of the Qualified Packets are described below and are listed above in Table 2. Once again, all packets start with a synchronization character chosen in the preferred embodiment to be 01 hex (01H). A Qualified Packet starts with two bytes where the first byte is the ID byte and the second byte is the QUALIFIER type identifier. Table 8 shows the ID byte for the Qualified Packet, Table 9 shows the byte structure of the Qualified Packet and Tables 10-12 list the Qualifier Type byte bit maps for the three types of Qualified Packets.

TABLE 8: ID Byte of Qualified Packet

	7	6	5	4	3	2	1	0
40	LI (Length Indicator) = 1 to 127							

The Length Identifier of the ID byte equals the amount of data which follows including the QUALIFIER byte (QUAL byte + DATA). If LI = 1, then the Qualifier Packet contains the Q byte only.

TABLE 9: Qualifier Packet Byte Structure

5	01 SYNC	85 ID	QUAL BYTE	data	data	data	data
---	------------	----------	--------------	------	------	------	------

The bit maps of the Qualifier Byte (QUAL BYTE) of the Qualified Packet are shown in Tables 10-12. The bit map follows the pattern whereby if the QUAL byte = 0, then the command is a break. Also, bit 1 of the QUAL byte designates ack/nak, bit 2 designates flow control and bit 6 designates stream mode command. Table 10 describes the Qualifier Byte of Qualified Packet, Group 1 which are immediate commands. Table 11 describes the Qualifier Byte of Qualified Packet, Group 2 which are stream mode commands in that the command is to stay in the designated mode until a BREAK + INIT command string is sent. Table 12 describes the Qualifier Byte of Qualified Packet, Group 3 which are information or status commands.

TABLE 10: Qualifier Byte of Qualified Packet: Group 1

20	7	6	5	4	3	2	1	0	
	x	x	x	x	x	x	x	x	
	0	0	0	0	0	0	0	0	= break
	0	0	0	0	0	0	1	0	= ACK
25	0	0	0	0	0	0	1	1	= NAK
	0	0	0	0	0	1	0	0	= xoff or stop sending data
	0	0	0	0	0	1	0	1	= xon or resume sending data
	0	0	0	0	1	0	0	0	= cancel fax

TABLE 11: Qualifier Byte of Qualified Packet: Group 2

35	7	6	5	4	3	2	1	0	
	x	x	x	x	x	x	x	x	
	0	1	0	0	0	0	0	1	= stream command mode
	0	1	0	0	0	0	1	0	= stream data
	0	1	0	0	0	0	1	1	= stream voice
	0	1	0	0	0	1	0	0	= stream video
	0	1	0	0	0	1	0	1	= stream A
40	0	1	0	0	0	1	1	0	= stream B
	0	1	0	0	0	1	1	1	= stream C

The Qualifier Packet indicating stream mode and BREAK attention is used when a large of amount of information is sent (voice, data...) to allow the highest throughput possible. This command is mainly intended

for use in DATA mode but can be used in any one of the possible modes. To change from one mode to another, a break-init sequence would be given. A break "AT...<cr>" type command would cause a change in state and set the serial rate from the "AT" command.

5 **TABLE 12: Qualifier Byte of Qualified Packet: Group 3**

	7	6	5	4	3	2	1	0	
	x	x	x	x	x	x	x	x	
10	1	0	0	0	0	0	0	0	= commands
	1	0	0	0	0	0	0	1	= responses
	1	0	0	0	0	0	1	0	= status

Cellular Supervisory Packet

15 In order to determine the status of the cellular link, a supervisory packet shown in Table 13 is used. Both sides of the cellular link will send the cellular supervisory packet every 3 seconds. Upon receiving the cellular supervisory packet, the receiving side will acknowledge it using the ACK field of the cellular supervisory packet. If the sender does not receive
 20 an acknowledgement within one second, it will repeat sending the cellular supervisory packet up to 12 times. After 12 attempts of sending the cellular supervisory packet without an acknowledgement, the sender will disconnect the line. Upon receiving an acknowledgement, the sender will restart its 3 second timer. Those skilled in the art will readily recognize that the timer
 25 values and wait times selected here may be varied without departing from the spirit or scope of the present invention.

30 **TABLE 13: Cellular Supervisory Packet Byte Structure**

8F	ID	LI	ACK	data	data	data
----	----	----	-----	------	------	------

Speech Compression

The Speech Compression algorithm described above for use in
 35 transmitting voice over data accomplished via the voice control circuit 306. Referring once again to Figure 2, the user is talking either through the handset, the headset or the microphone/speaker telephone interface. The analog voice signals are received and digitized by the telephone CODEC circuit 305. The digitized voice information is passed from the digital

telephone CODEC circuit 305 to the voice control circuits 306. The digital signal processor (DSP) of the voice control circuit 306 is programmed to do the voice compression algorithm. The source code programmed into the voice control DSP is in the microfiche appendix of U.S. Patent Application Serial

5 No. 08/002,467, filed January 8, 1993 entitled "COMPUTER-BASED MULTIFUNCTION PERSONAL COMMUNICATIONS SYSTEM". The DSP of the voice control circuit 306 compresses the speech and places the compressed digital representations of the speech into special packets described more fully below. As a result of the voice compression algorithm, the

10 compressed voice information is passed to the dual port ram circuit 308 for either forwarding and storage on the disk of the personal computer via the RS232 serial interface or for multiplexing with conventional modem data to be transmitted over the telephone line via the telephone line interface circuit 309 in the voice-over-data mode of operation Show and Tell function 123.

15 The compressed speech bits are multiplexed with data bits using a packet format described below. Three compression rates are described herein which will be called 8Kbit/sec, 9.6Kbit/sec and 16Kbit/sec.

Speech Compression Algorithm

To multiplex high-fidelity speech with digital data and transmit

20 both over the telephone line, a high available bandwidth would normally be required. In the present invention, the analog voice information is digitized into 8-bit PCM data at an 8 KHz sampling rate producing a serial bit stream of 64,000 bps serial data rate. This rate cannot be transmitted over the telephone line. With the Speech Compression algorithm described below, the

25 64 Kbs digital voice data is compressed into a 9500 bps encoding bit stream using a fixed-point (non-floating point) DSP such that the compressed speech can be transmitted over the telephone line multiplexed with asynchronous data. This is accomplished in an efficient manner such that enough machine cycles remain during real time speech compression to allow for echo

30 cancellation in the same fixed-point DSP.

A silence detection function is used to detect quiet intervals in the speech signal which allows the data processor to substitute asynchronous

data in lieu of voice data packets over the telephone line to efficiently time multiplex the voice and asynchronous data transmission. The allocation of time for asynchronous data transmission is constantly changing depending on how much silence is on the voice channel.

5 The voice compression algorithm of the present system relies on a model of human speech which shows that human speech contains redundancy inherent in the voice patterns. Only the incremental innovations (changes) need to be transmitted. The algorithm operates on 128 digitized speech samples (20 milliseconds at 6400 Hz), divides the speech samples into
10 time segments of 32 samples (5 milliseconds) each, and uses predicted coding on each segment. Thus, the input to the algorithm could be either PCM data sampled at 6400 Hz or 8000 Hz. If the sampling is at 8000 Hz, or any other selected sampling rate, the input sample data stream must be decimated from
15 6400 Hz PCM signal is interpolated back to 8000 Hz and passed to the CODEC.

 With this algorithm, the current segment is predicted as best as possible based on the past recreated segments and a difference signal is determined. The difference values are compared to the stored difference
20 values in a lookup table or code book, and the address of the closest value is sent to the remote site along with the predicted gain and pitch values for each segment. In this fashion, the entire 20 milliseconds of speech can be represented by 190 bits, thus achieving an effective data rate of 9500 bps.

 To produce this compression, the present system includes a
25 unique Vector Quantization (VQ) speech compression algorithm designed to provide maximum fidelity with minimum compute power and bandwidth. The VQ algorithm has two major components. The first section reduces the dynamic range of the input speech signal by removing short term and long term redundancies. This reduction is done in the waveform domain, with the
30 synthesized part used as the reference for determining the incremental "new" content. The second section maps the residual signal into a code book optimized for preserving the general spectral shape of the speech signal.

Figure 3 is a high level signal flow block diagram of the speech compression algorithm used in the present system to compress the digitized voice for transmission over the telephone line in the voice over data mode of operation or for storage and use on the personal computer. The transmitter and receiver components are implemented using the programmable voice control DSP/CODEC circuit 306 shown in Figure 2.

The DC removal stage 1101 receives the digitized speech signal and removes the D.C. bias by calculating the long-term average and subtracting it from each sample. This ensures that the digital samples of the speech are centered about a zero mean value. The pre-emphasis stage 1103 whitens the spectral content of the speech signal by balancing the extra energy in the low band with the reduced energy in the high band.

The system finds the innovation in the current speech segment by subtracting 1109 the prediction from reconstructed past samples synthesized from synthesis stage 1107. This process requires the synthesis of the past speech samples locally (analysis by synthesis). The synthesis block 1107 at the transmitter performs the same function as the synthesis block 1113 at the receiver. When the reconstructed previous segment of speech is subtracted from the present segment (before prediction), a difference term is produced in the form of an error signal. This residual error is used to find the best match in the code book 1105. The code book 1105 quantizes the error signal using a code book generated from a representative set of speakers and environments. A minimum mean squared error match is determined in segments. In addition, the code book is designed to provide a quantization error with spectral rolloff (higher quantization error for low frequencies and lower quantization error for higher frequencies). Thus, the quantization noise spectrum in the reconstructed signal will always tend to be smaller than the underlying speech signal.

The following description will specifically explain the algorithm for the 9.6Kbit/sec compression rate, except where specifically stated otherwise. The discussion is applicable to the other compression rates by substituting the parameter values found in Table 14, below, and by following

the special instructions for each calculation provided throughout the discussion.

**TABLE 14 Speech Compression Algorithm Parameters For
Three Voice Compression Rates**

	<u>Parameter</u>	<u>16Kbit/sec</u>	<u>9.6Kbit/sec</u>	<u>8Kbit/sec</u>
5	Input Samples (msecs @ 8Ksample/sec)	160 (20msec)	160 (20msec)	192 (24msec)
10	Decimated Samples (msec) @ rate	160 (20msec) @ 8Ksample/sec	128 (20msec) @ 6.4Ksample/sec	144 (24msec) @ 6Ksample/sec
	Sub-Block Size	40	32	36
15	Min_Pitch	40	32	36
	Max_Pitch	160	95	99
	Codebook Size	256	512	512
20	Vector Size (VSIZE)	5	8	9
	# of compressed bytes	40	24	24
25	ρ	0.75	0.5	0.5

9.6Kbit/sec Compression Rate Algorithm

For the 9.6Kbit/sec speech compression rate, each frame of 20ms is divided into 4 sub-blocks or segments of 5ms each. Each sub-block of data consists of a plurality of bits for the long term predictor, a plurality of bits for the long term predictor gain, a plurality of bits for the sub-block gain, and a plurality of bits for each code book entry for each 5ms. In the code book block, each 1.25ms of speech is looked up in a 512 word code book for the best match. The table entry is transmitted rather than the actual samples. The code book entries are pre-computed from representative speech segments, as described more fully below.

On the receiving end 1200, the synthesis block 1113 at the receiver performs the same function as the synthesis block 1107 at the transmitter. The synthesis block 1113 reconstructs the original signal from the voice data packets by using the gain and pitch values and code book address corresponding to the error signal most closely matched in the code book. The code book at the receiver is similar to the code book 1105 in the transmitter. Thus the synthesis block recreates the original pre-emphasized signal. The de-emphasis stage 1115 inverts the pre-emphasis operation by restoring the balance of original speech signal.

The complete speech compression algorithm is summarized as follows:

- a) Digitally sample the voice to produce a PCM sample bit stream sampled at 8,000 samples per second.
- b) Decimate the 8,000 samples per second sampled data to produce a sampling rate of 6,400 samples per second for the 9.6Kbit/sec compression rate (6,000 samples per second for the 8Kbit/sec algorithm and 8,000 samples per second for the 16Kbit/sec algorithm).
- c) Remove any D.C. bias in the speech signal.
- d) Pre-emphasize the signal.
- e) Find the innovation in the current speech segment by subtracting the prediction from reconstructed past samples. This step requires the synthesis of the past

speech samples locally (analysis by synthesis) such that the residual error is fed back into the system.

- 5 f) Quantize the error signal using a code book generated from a representative set of speakers and environments. A minimum mean squared error match is determined in 5ms segments. In addition, the code book is designed to provide a quantization error with spectral rolloff (higher quantization error for low frequencies and lower
- 10 quantization error for higher frequencies). Thus, the quantization noise spectrum in the reconstructed signal will always tend to be smaller than the underlying speech signal.
- 15 g) At the transmitter and the receiver, reconstruct the speech from the quantized error signal fed into the inverse of the function in step (e) above. Use this signal for analysis by synthesis and for the output to the reconstruction stage below.
- 20 h) Use a de-emphasis filter to reconstruct the output.

The major advantages of this approach over other low-bit-rate algorithms are that there is no need for any complicated calculation of reflection coefficients (no matrix inverse or lattice filter computations). Also, the quantization noise in the output speech is hidden under the speech signal and there are no pitch tracking artifacts: the speech sounds "natural", with only minor increases of background hiss at lower bit-rates. The computational load is reduced significantly compared to a VSELP algorithm and variations of the present algorithm thus provides bit rates of 8, 9.6 and 16 Kbit/sec, and can also provide bit rates of 9.2Kbit/sec, 9.5Kbit/sec and many other rates. The total delay through the analysis section is less than 20 milliseconds in the 9.6Kbit/sec embodiment. The present algorithm is accomplished completely in the waveform domain and there is no spectral information being computed

35 and there is no filter computations needed.

Detailed Description of the Speech Compression Algorithm

The speech compression algorithm is described in greater detail with reference to Figures 4 through 7, and with reference to the block diagram of the hardware components of the present system shown at Figure 2. The voice compression algorithm operates within the programmed control of the

40

voice control DSP circuit 306. In operation, the speech or analog voice signal is received through the telephone interface 301, 302 or 303 and is digitized by the digital telephone CODEC circuit 305. The CODEC for circuit 305 is a companding μ -law CODEC. The analog voice signal from the telephone interface is band-limited to about 3,000 Hz and sampled at a selected sampling rate by digital telephone CODEC 305. The sample rates in the 9.6Kbit/sec embodiment of the present invention are 8Ksample/sec. Each sample is encoded into 8-bit PCM data producing a serial 64kb/s. The digitized samples are passed to the voice control DSP/CODEC of circuit 306.

There, the 8-bit μ -law PCM data is converted to 13-bit linear PCM data. The 13-bit representation is necessary to accurately represent the linear version of the logarithmic 8-bit μ -law PCM data. With linear PCM data, simpler mathematics may be performed on the PCM data.

The voice control DSP/CODEC of circuit 306 correspond to the single integrated circuit U8 shown in Figures 9A and 9B as a *WE[®]* DSP16C Digital Signal Processor/CODEC from AT&T Microelectronics which is a combined digital signal processor and a linear CODEC in a single chip as described above. The digital telephone CODEC of circuit 305 corresponds to integrated circuit U12 shown in Figure 9B as a T7540 companding μ -law CODEC.

The sampled and digitized PCM voice signals from the telephone μ -law CODEC 305 shown in Figure 2 are passed to the voice control DSP/CODEC circuit 308 via direct data lines clocked and synchronized to a clocking frequency. The sample rate in CODEC 305 in this embodiment of the present invention is 8Ksample/sec. The digital samples are loaded into the voice control DSP/CODEC one at a time through the serial input and stored into an internal queue held in RAM, converted to linear PCM data and decimated to a sample rate of 6.4Ksample/sec. As the samples are loaded into the end of the queue in the RAM of the voice control DSP, the samples at the head of the queue are operated upon by the voice compression algorithm. The voice compression algorithm then produces a greatly compressed representation of the speech signals in a digital packet form. The

compressed speech signal packets are then passed to the dual port RAM circuit 308 shown in Figure 2 for use by the main controller circuit 313 for either transferring in the voice-over-data mode of operation or for transfer to the personal computer for storage as compressed voice for functions such as telephone answering machine message data, for use in the multi-media documents and the like.

In the voice-over-data mode of operation, voice control DSP/CODEC circuit 306 of Figure 2 will be receiving digital voice PCM data from the digital telephone CODEC circuit 305, compressing it and transferring it to dual port RAM circuit 308 for multiplexing and transfer over the telephone line. This is the transmit mode of operation of the voice control DSP/CODEC circuit 306 corresponding to transmitter block 1100 of Figure 3 and corresponding to the compression algorithm of Figure 4.

Concurrent with this transmit operation, the voice control DSP/CODEC circuit 306 is receiving compressed voice data packets from dual port RAM circuit 308, uncompressing the voice data and transferring the uncompressed and reconstructed digital PCM voice data to the digital telephone CODEC 305 for digital to analog conversion and eventual transfer to the user through the telephone interface 301, 302, 304. This is the receive mode of operation of the voice control DSP/CODEC circuit 306 corresponding to receiver block 1200 of Figure 3 and corresponding to the decompression algorithm of Figure 5. Thus, the voice-control DSP/CODEC circuit 306 is processing the voice data in both directions in a full-duplex fashion.

The voice control DSP/CODEC circuit 306 operates at a clock frequency of approximately 24.576MHz while processing data at sampling rates of approximately 8KHz in both directions. The voice compression/decompression algorithms and packetization of the voice data is accomplished in a quick and efficient fashion to ensure that all processing is done in real-time without loss of voice information. This is accomplished in an efficient manner such that enough machine cycles remain in the voice

control DSP circuit 306 during real time speech compression to allow real time acoustic and line echo cancellation in the same fixed-point DSP.

In programmed operation, the availability of an eight-bit sample of PCM voice data from the μ -law digital telephone CODEC circuit 305 causes an interrupt in the voice control DSP/CODEC circuit 306 where the sample is loaded into internal registers for processing. Once loaded into an internal register it is transferred to a RAM address which holds a queue of samples. The queued PCM digital voice samples are converted from 8-bit μ -law data to a 13-bit linear data format using table lookup for the conversion. Those skilled in the art will readily recognize that the digital telephone CODEC circuit 305 could also be a linear CODEC.

Sample Rate Decimation

The sampled and digitized PCM voice signals from the telephone μ -law CODEC 305 shown in Figure 2 are passed to the voice control DSP/CODEC circuit 308 via direct data lines clocked and synchronized to a clocking frequency. The sample rate in this embodiment of the present invention is 8Ksample/sec. The digital samples for the 9.6Kbit/sec and 8Kbit/sec algorithms are decimated using a digital decimation process to produce a 6.4Ksample/sec and 6Ksample/sec rate, respectively. For the 16Kbit/sec algorithm, no decimation is needed.

Referring to Figure 3, the decimated digital samples are shown as speech entering the transmitter block 1100. The transmitter block, of course, is the mode of operation of the voice-control DSP/CODEC circuit 306 operating to receive local digitized voice information, compress it and packetize it for transfer to the main controller circuit 313 for transmission on the telephone line. The telephone line connected to telephone line interface 309 of Figure 2 corresponds to the channel 1111 of Figure 3.

A frame rate for the voice compression algorithm is 20 milliseconds of speech for each compression. This correlates to 128 samples to process per frame for the 6.4K decimated sampling rate. When 128 samples are accumulated in the queue of the internal DSP RAM, the compression of that sample frame is begun.

Data Flow Description

The voice-control DSP/CODEC circuit 306 is programmed to first remove the DC component 1101 of the incoming speech. The DC removal is an adaptive function to establish a center base line on the voice
 5 signal by digitally adjusting the values of the PCM data. This corresponds to the DC removal stage 1203 of the software flow chart of Figure 4. The formula for removal of the DC bias or drift is as follows:

$$10 \quad x(n) = s(n) - s(n-1) + \alpha * x(n-1) \quad \text{where } \alpha = \frac{32735}{32768}$$

and where n = sample number,
 $s(n)$ is the current sample, and
 15 $x(n)$ is the sample with the DC bias removed.

The removal of the DC is for the 20 millisecond frame of voice which amounts to 128 samples at the 6.4Ksample/sec decimated sampling rate which corresponds to the 9.6Kbit/sec algorithm. The selection of α is based
 20 on empirical observation to provide the best result.

Referring again to Figure 4, the voice compression algorithm in a control flow diagram is shown which will assist in the understanding of the block diagram of Figure 3. Figure 6 is a simplified data flow description of the flow chart of Figure 4 showing the sample rate decimator 1241 and the
 25 sample rate incrementor 1242. Sample rate decimator 1241 produces an output 1251 of 6.4Ksample/sec for an 8Ksample/sec input in the 9.6Kbit/sec system. (Similarly, a 6Ksample/sec output 1250 is produced for the 8Kbit/sec algorithm, and no decimation is performed on the 8Ksample/sec voice sample rate 1252 for the 16Kbit/sec algorithm.) The analysis and compression begin
 30 at block 1201 where the 13-bit linear PCM speech samples are accumulated until 128 samples (for the 6.4Ksample/sec decimated sampling rate) representing 20 milliseconds of voice or one frame of voice is passed to the DC removal portion of code operating within the programmed voice control DSP/CODEC circuit 306. The DC removal portion of the code described

above approximates the base line of the frame of voice by using an adaptive DC removal technique.

A silence detection algorithm 1205 is also included in the programmed code of the DSP/CODEC 306. The silence detection function is a summation of the square of each sample of the voice signal over the frame. If the power of the voice frame falls below a preselected threshold, this would indicate a silent frame. The detection of a silence frame of speech is important for later multiplexing of the V-data (voice data) and C-data (asynchronous computer data) described below. During silent portions of the speech, the main controller circuit 313 will transfer conventional digital data (C-data) over the telephone line in lieu of voice data (V-data). The formula for computing the power is

$$PWR = \sum_{n=0}^{\text{Sub-Block Size}-1} x(n) * x(n)$$

where n is the sample number, and
x (n) is the sample value

If the power PWR is lower than a preselected threshold, then the present voice frame is flagged as containing silence. The 128-sample (Decimated Samples) silent frame is still processed by the voice compression algorithm; however, the silent frame packets are discarded by the main controller circuit 313 so that asynchronous digital data may be transferred in lieu of voice data. The rest of the voice compression is operated upon in segments where there are four segments per frame amounting to 32 samples of data per segment (Sub-Block Size). It is only the DC removal and silence detection which is accomplished over an entire 20 millisecond frame.

The pre-emphasis 1207 of the voice compression algorithm shown in Figure 4 is the next step. The sub-blocks are first passed through a pre-emphasis stage which whitens the spectral content of the speech signal by balancing the extra energy in the low band with the reduced energy in the high band. The pre-emphasis essentially flattens the signal by reducing the

dynamic range of the signal. By using pre-emphasis to flatten the dynamic range of the signal, less of a signal range is required for compression making the compression algorithm operate more efficiently. The formula for the pre-emphasis is

- 5 $x(n) = x(n) - \rho * x(n-1)$ where $\rho = 0.5$ for 9.6Kbit/sec
 and where n is the sample number,
 $x(n)$ is the sample

- Each segment thus amounts to five milliseconds of voice which is equal to 32 samples. Pre-emphasis then is done on each segment. The
10 selection of ρ is based on empirical observation to provide the best result.

- The next step is the long-term prediction (LTP). The long-term prediction is a method to detect the innovation in the voice signal. Since the voice signal contains many redundant voice segments, we can detect these redundancies and only send information about the changes in the signal from
15 one segment to the next. This is accomplished by comparing the speech samples of the current segment on a sample by sample basis to the reconstructed speech samples from the previous segments to obtain the innovation information and an indicator of the error in the prediction.

- The long-term predictor gives the pitch and the LTP-Gain of
20 the sub-block which are encoded in the transmitted bit stream. In order to predict the pitch in the current segment, we need at least 3 past sub-blocks of reconstructed speech. This gives a pitch value in the range of MIN_PITCH to MAX_PITCH (32 and 95, respectively, as given in Table 14). This value is coded with 6-bits. But, in order to accommodate the compressed data rate
25 within a 9600 bps link, the pitch for segments 0 and 3 is encoded with 6 bits, while the pitch for segments 1 and 2 is encoded with 5 bits. When performing the prediction of the Pitch for segments 1 and 2, the correlation lag is adjusted around the predicted pitch value of the previous segment. This gives us a good chance of predicting the correct pitch for the current segment
30 even though the entire range for prediction is not used. The computations for the long-term correlation lag PITCH and associated LTP-gain factor β_j

(where $j = 0, 1, 2, 3$ corresponding to each of the four segments of the frame) are done as follows:

For $j = \text{min_pitch} \dots \text{max_pitch}$, first perform the following computations between the current speech samples $x(n)$ and the past

5 reconstructed speech samples $x'(n)$

Sub-Block Size - 1

$$S_{xx'}(j) = \sum_{i=0}^{\text{Sub-Block Size} - 1} x(i) * x'(i + \text{MAX_PITCH} - j)$$

10

$$S_{xx'}(j) = \sum_{i=0}^{\text{Sub-Block Size} - 1} x'(i + \text{MAX_PITCH} - j) * x'(i + \text{MAX_PITCH} - j)$$

15

The Pitch j is chosen as that which maximizes $\frac{S_{xx'}^2}{S_{xx'}}$. Since βj is positive, only j with positive $S_{xx'}^2$ is considered.

For the 9.6Kbit/sec and 8Kbit/sec embodiments, the Pitch is encoded with different number of bits for each sub-segment, the value of

20 min_pitch and max_pitch (range of the synthesized speech for pitch prediction of the current segment) is computed as follows:

if (seg_number = 0 or 3)

{

min_pitch = MIN_PITCH

25

max_pitch = MAX_PITCH

}

if (seg_number = 1 or 2)

{

min_pitch = prev_pitch - 15

30

if (prev_pitch < MIN_PITCH + 15)

min_pitch = MIN_PITCH

if (prev_pitch > MAX_PITCH + 15)

min_pitch = MAX_PITCH - 30

max_pitch = min_pitch + 30

35

}

(This calculation is not necessary for the 16Kbit/sec algorithm.)

The **prev_pitch** parameter in the above equation, is the of the pitch of the previous sub-segment. The pitch j is the encoded in 6 bits or 5 bits as:

$$\text{encoded bits} = j - \text{min_pitch}$$

5 The LTP-Gain is given by

$$\beta = \frac{S_{xx'}(j)}{S_{xx}(j)} \quad \text{for } S_{xx}(j) \neq 0$$

10 The value of the β is a normalized quantity between zero and unity for this segment where β is an indicator of the correlation between the segments. For example, a perfect sine wave would produce a β which would be close to unity since the correlation between the current segments and the previous reconstructed segments should be almost a perfect match so β is one.

15 The LTP gain factor is quantized from a LTP Gain Encode Table. This table is characterized in Table 15. The resulting index (**bcode**) is transmitted to the far end. At the receiver, the LTP Gain Factor is retrieved from Table 16, as follows:

$$\beta_q = \text{dlb_tab}[\text{bcode}]$$

20 **TABLE 15: LTP Gain Encode Table**

$\beta =$	0.1	0.3	0.5	0.7	0.9	
\leftarrow						\rightarrow
bcode= 0	1	2	3	4	5	

25 **TABLE 16: LTP Gain Decode Table**

$\beta =$	0.0	0.2	0.4	0.5	0.8	1.0
\leftarrow						\rightarrow
30 bcode=0	1	2	3	4	5	

30

After the Long-Term Prediction, we pass the signal through a pitch filter to whiten the signal so that all the pitch effects are removed. The pitch filter is given by:

35

38

$$e(n) = x(n) - \beta_q * x'(n-j)$$

where j is the Lag, and

β_q is the associated Quantized LTP Gain.

Next, the error signal is normalized with respect to the

- 5 maximum amplitude in the sub-segment for vector-quantization of the error signal. The maximum amplitude in the segment is obtained as follows:

$$G = \text{MAX}\{|e(n)|\}$$

The maximum amplitude (G) is encoded using the Gain Encode Table. This table is characterized in Table 17. The encoded amplitude

- 10 (**gcode**) is transmitted to the far end. At the receiver, the maximum amplitude is retrieved from Table 18, as follows:

$$G_q = \text{dlg_tab}[\text{gcode}]$$

The error signal $e(n)$ is then normalized by

$$15 \quad e(n) = \frac{e(n)}{G_q}$$

TABLE 17: Gain Encode Table

G=16	32	64	128	256	512	1024	2048	4096	8192
<---	---	---	---	---	---	---	---	---	---
0	1	2	3	4	5	6	7	8	9
(gcode)									

TABLE 18: Gain Decode Table

G=16	32	64	128	256	512	1024	2048	4096	8192
---	---	---	---	---	---	---	---	---	---
0	1	2	3	4	5	6	7	8	9
(gcode)									

- From the Gain and LTP Gain Encode tables, we can see that
- 30 we would require 4 bits for **gcode** and 3 bits for **bcode**. This results in total of 7 bits for both parameters. In order to reduce the bandwidth of the compressed bit stream, the **gcode** and **bcode** parameters are encoded together in 6 bits, as follows:

$$\text{BGCODE} = 6 * \text{gcode} + \text{bcode}.$$

35

The encoded bits for the G and LTP-Gain (β) at the receiver can be obtained as follows:

$$\text{gcode} = \text{BGCODE} / 6$$

$$\text{bcode} = \text{BGCODE} - 6 * \text{gcode}$$

- 5 However, these calculations are needed only for the 8Kbit/sec and 9.6Kbit/sec algorithms.

Each segment of 32 samples (Sub-Block Size) is divided into 4 vectors of 8 samples (VSIZE) each. Each vector is compared to the vectors stored in the **CodeBook** and the **Index** of the **Code Vector** that is closest to the
 10 signal vector is selected. The **CodeBook** consists of 512 entries (512 addresses). The index chosen has the least difference according to the following minimization formula:

$$\text{Min} \left\{ \sum_{i=0}^{\text{VSIZE}-1} (x_i - y_i)^2 \right\}$$

- 15 where x_i = the input vector of VSIZE samples (8 for the 9.6Kbit/sec algorithm), and
 y_i = the code book vector of VSIZE samples (8 for the 9.6Kbit/sec algorithm).

- 20 The minimization computation, to find the best match between the subsegment and the code book entries is computationally intensive. A brute force comparison may exceed the available machine cycles if real time processing is to be accomplished. Thus, some shorthand processing approaches are taken to reduce the computations required to find the best fit.
 25 The above formula can be computed in a shorthand fashion as follows.

By expanding out the above formula, some of the unnecessary terms may be removed and some fixed terms may be pre-computed:

$$\begin{aligned} (x_i - y_i)^2 &= (x_i - y_i) * (x_i - y_i) \\ &= (x_i^2 - x_i y_i - x_i y_i + y_i^2) \\ &= (x_i^2 - 2x_i y_i + y_i^2) \end{aligned}$$

- 30 where x_i^2 is a constant so it may be dropped from the formula,

and the value of $-\frac{1}{2} \sum y_i^2$ may be precomputed and stored as the $VSIZE + 1$ th value ($8 + 1 = 9$ th value for the 9.6 Kbit/sec algorithm) in the code book so that the only real-time computation involved is the following formula:

$$5 \quad \text{Min} \left\{ \sum_{i=0}^{VSIZE-1} (x_i - y_{ii}) \right\}$$

Thus, for a segment of Sub-Block Size samples (32 for the 9.6Kbit/sec algorithm), we will transmit Sub-Block Size/VSIZE CodeBook indices (4 CodeBook Indices, 9 bits each, for the 9.6Kbit/sec algorithm).
10 Therefore, for the 9.6Kbit/sec algorithm, for each Sub-Block Size segment we will transmit 36 bits representing that segment.

After the appropriate index into the code book is chosen, the input speech samples are replaced by the corresponding vectors in the chosen
15 indexes. These values are then multiplied by the G_q to denormalize the synthesized error signal, $e'(n)$. This signal is then passed through the Inverse Pitch Filter to reintroduce the Pitch effects that was taken out by the Pitch filter. The Inverse Pitch Filter is performed as follows:

$$y(n) = e'(n) + \beta_q * x'(n - j)$$

20 where β_q is the Quantized LTP-Gain from Table 16, and j is the Lag.

The Inverse Pitch Filter output is used to update the synthesized speech buffer which is used for the analysis of the next sub-segment. The update of the state buffer is as follows:

$$x'(k) = x'(k + MIN_PITCH)$$

25 where $k = 0, \dots, (MAX_PITCH - MIN_PITCH) - 1$

$$x'(l) = y(n)$$

where $l = MAX_PITCH - MIN_PITCH, \dots, MAX_PITCH - 1$

The signal is then passed through the deemphasis filter since preemphasis was performed at the beginning of the processing. In the
30 analysis, only the preemphasis state is updated so that we properly satisfy the Analysis-by-Synthesis method of performing the compression. In the Synthesis, the output of the deemphasis filter, $s'(n)$, is passed on to the D/A to generate analog speech. The deemphasis filter is implemented as follows:

$s'(n) = y(n) + \rho * s'(n - 1)$ where $\rho = 0.5$ for the 9.6Kbit/sec algorithm

The voice is reconstructed at the receiving end of the voice-over data link according to the reverse of the compression algorithm as shown
5 as the decompression algorithm in Figure 5.

If a silence frame is received, the decompression algorithm simply discards the received frame and initialize the output with zeros. If a speech frame is received, the pitch, LTP-Gain and GAIN are decoded as explained above. The error signal is reconstructed from the codebook
10 indexes, which is then denormalized with respect to the GAIN value. This signal is then passed through the Inverse filter to generate the reconstructed signal. The Pitch and the LTP-Gain are the decoded values, same as those used in the Analysis. The filtered signal is passed through the Deemphasis filter whose output is passed on to the D/A to put out analog speech.

15 The compressed frame contains 23 8-bit words and one 6-bit word. Thus a total of 24 words. Total number of bits transferred is 190, which corresponds to 9500 bps as shown in Table 19 (for the 9.6Kbit/sec algorithm).

Table 19 Compressed Frame Packet for 9.6Kbit/sec Algorithm

	7	6	5	4	3	2	1	0	Bit Number
	S	S	p_0^5	p_0^4	p_0^3	p_0^2	p_0^1	p_0^0	Comp_Frame[0]
	V_2^8	V_1^8	V_0^8	p_1^4	p_1^3	p_1^2	p_1^1	p_1^0	Comp_Frame[1]
5	V_5^8	V_4^8	V_3^8	p_2^4	p_2^3	p_2^2	p_2^1	p_2^0	Comp_Frame[2]
	V_7^8	V_6^8	p_3^5	p_3^4	p_3^3	p_3^2	p_3^1	p_3^0	Comp_Frame[3]
	V_9^8	V_8^8	BG_0^5	BG_0^4	BG_0^3	BG_0^2	BG_0^1	BG_0^0	Comp_Frame[4]
	V_{11}^8	V_{10}^8	BG_1^5	BG_1^4	BG_1^3	BG_1^2	BG_1^1	BG_1^0	Comp_Frame[5]
	V_{13}^8	V_{12}^8	BG_2^5	BG_2^4	BG_2^3	BG_2^2	BG_2^1	BG_2^0	Comp_Frame[6]
10	V_{15}^8	V_{14}^8	BG_3^5	BG_3^4	BG_3^3	BG_3^2	BG_3^1	BG_3^0	Comp_Frame[7]
	VQ_0^7	VQ_0^6	VQ_0^5	VQ_0^4	VQ_0^3	VQ_0^2	VQ_0^1	VQ_0^0	Comp_Frame[8] = LS 8 bits VQ[0]
	VQ_1^7	VQ_1^6	VQ_1^5	VQ_1^4	VQ_1^3	VQ_1^2	VQ_1^1	VQ_1^0	Comp_Frame[9] = LS 8 bits VQ[1]
15	VQ_{14}^7	VQ_{14}^6	VQ_{14}^5	VQ_{14}^4	VQ_{14}^3	VQ_{14}^2	VQ_{14}^1	VQ_{14}^0	Comp_Frame[22] LS 8 bits VQ[14]
	VQ_{15}^7	VQ_{15}^6	VQ_{15}^5	VQ_{15}^4	VQ_{15}^3	VQ_{15}^2	VQ_{15}^1	VQ_{15}^0	Comp_Frame[23] LS 8 bits VQ[15]

where BG = Beta/Gain, P = Pitch, VQ = CodeBook Index and S = Spare

20 Bits

Code Book Descriptions

Table 20 describes the format of the code book for the 9.6Kbit/sec algorithm. The code book values are stored in a signed floating point format which is converted to a fixed point representation of floating point number

25

when stored in the lookup tables of the present invention. There are 512 entries in each code book corresponding to 512 different speech segments which can be used to encode and reconstruct the speech.

Table 20: Code Book Format for the 9.6Kbit/sec Algorithm

5

—————	Code Book Entries	—————	— $\frac{1}{2} \text{Sum}^2$ Constant —
	8 entries		1 entry

For the 9.6Kbit/sec algorithm, the code book comprises a table of nine columns and 512 rows of floating point data. The first 8 rows
 10 correspond to the 8 samples of speech and the ninth entry is the precomputed constant described above as $-\frac{1}{2} \sum y_i^2$. An example of the code book data is shown in Table 21 with the complete code book for the 9.6Kbit/sec algorithm.

Table 21: Code Book Example for the 2.6Kbit/sec Algorithm

0.786438	1.132875	1.208375	1.206750	1.114250	0.937688	0.772062	0.583250	3.93769
5 0.609667	1.019167	0.909167	0.957750	0.999833	0.854333	1.005667	0.911250	3.36278
0.614750	1.150750	1.477750	1.548750	1.434750	1.304250	1.349750	1.428250	6.95291
0.657000	1.132909	1.279909	1.204727	1.335636	1.280818	1.162000	0.958818	5.24933
0.592429	0.897571	1.101714	1.337286	1.323571	1.349000	1.304857	1.347143	5.6239
0.325909	0.774182	1.035727	1.263636	1.456455	1.356273	1.076273	0.872818	4.628

- The code books are stored in PROM memory accessible by the Voice DSP as a lookup table. The table data is loaded into local DSP
- 5 memory upon the selection of the appropriate algorithm to increase access speed. The code books comprise a table of data in which each entry is a sequential address from 000 to 511. For the 9.6Kbit/sec algorithm, a 9 X 512 code book is used. For the 16Kbit/sec algorithm, a 6 X 256 code book is used and for the 8Kbit/sec algorithm, a 9 X 512 code book is used.
- 10 Depending upon which voice compression quality and compression rate is selected, the corresponding code book is used to encode/decode the speech samples.

Generation of the Code Books

- The code books are generated statistically by encoding a wide
- 15 variety of speech patterns. The code books are generated in a learning mode for the above-described algorithm in which each speech segment which the compression algorithm is first exposed to is placed in the code book until 512 entries are recorded. Then the algorithm is continually fed a variety of speech patterns upon which the code book is adjusted. As new speech segments are
- 20 encountered, the code book is searched to find the best match. If the error between the observed speech segment and the code book values exceed a predetermined threshold, then the closest speech segment in the code book and the new speech segment is averaged and the new average is placed in the code book in place of the closest match. In this learning mode, the code book
- 25 is continually adjusted to have the lowest difference ratio between observed speech segment values and code book values. The learning mode of operation may take hours or days of exposure to different speech patterns to adjust the code books to the best fit.

- The code books may be exposed to a single person's speech
- 30 which will result in a code book being tailored to that particular persons method of speaking. For a mass market sale of this product, the speech patterns of a wide variety of speakers of both genders are exposed to the code

book learning algorithm for the average fit for a given language. For other languages, it is best to expose the algorithm to speech patterns of only one language such as English or Japanese.

Voice Over Data Packet Protocol

5 As described above, the present system can transmit voice data and conventional data concurrently by using time multiplex technology. The digitized voice data, called V-data carries the speech information. The conventional data is referred to as C-data. The V-data and C-data multiplex transmission is achieved in two modes at two levels: the transmit and receive
10 modes and data service level and multiplex control level. This operation is shown diagrammatically in Figure 7.

 In transmit mode, the main controller circuit 313 of Figure 2 operates in the data service level 1505 to collect and buffer data from both the personal computer 10 (through the RS232 port interface 315) and the voice
15 control DSP 306. In multiplex control level 1515, the main controller circuit 313 multiplexes the data and transmits that data out over the phone line 1523. In the receive mode, the main controller circuit 313 operates in the multiplex control level 1515 to de-multiplex the V-data packets and the C-data packets and then operates in the data service level 1505 to deliver the appropriate data
20 packets to the correct destination: the personal computer 10 for the C-data packets or the voice control DSP circuit 306 for V-data.

Transmit Mode

 In transmit mode, there are two data buffers, the V-data buffer 1511 and the C-data buffer 1513, implemented in the main controller RAM
25 316 and maintained by main controller 313. When the voice control DSP circuit 306 engages voice operation, it will send a block of V-data every 20 ms to the main controller circuit 313 through dual port RAM circuit 308. Each V-data block has one sign byte as a header and 24 bytes of V-data.

 The sign byte header of the voice packet is transferred every
30 frame from the voice control DSP to the controller 313. The sign byte header contains the sign byte which identifies the contents of the voice packet. The sign byte is defined as follows:

00 hex = the following V-data contains silent sound

01 hex = the following V-data contains speech information

If the main controller 313 is in transmit mode for V-data/C-data multiplexing, the main controller circuit 313 operates at the data service level to perform the following tests. When the voice control DSP circuit 306 starts to send the 24-byte V-data packet through the dual port RAM to the main controller circuit 313, the main controller will check the V-data buffer to see if the buffer has room for 24 bytes. If there is sufficient room in the V-data buffer, the main controller will check the sign byte in the header preceding the V-data packet. If the sign byte is equal to one (indicating voice information in the packet), the main controller circuit 313 will put the following 24 bytes of V-data into the V-data buffer and clear the silence counter to zero. Then the main controller 313 sets a flag to request that the V-data be sent by the main controller at the multiplex control level.

If the sign byte is equal to zero (indicating silence in the V-data packet), the main controller circuit 313 will increase the silence counter by 1 and check if the silence counter has reached 5. When the silence counter reaches 5, the main controller circuit 313 will not put the following 24 bytes of V-data into the V-data buffer and will stop increasing the silence counter. By this method, the main controller circuit 313 operating at the service level will only provide non-silence V-data to the multiplex control level, while discarding silence V-data packets and preventing the V-data buffer from being overwritten.

The operation of the main controller circuit 313 in the multiplex control level is to multiplex the V-data and C-data packets and transmit them through the same channel. At this control level, both types of data packets are transmitted by the HDLC protocol in which data is transmitted in synchronous mode and checked by CRC error checking. If a V-data packet is received at the remote end with a bad CRC, it is discarded since 100% accuracy of the voice channel is not ensured. If the V-data packets were re-sent in the event of corruption, the real-time quality of the

voice transmission would be lost. In addition, the C-data is transmitted following a modem data communication protocol such as CCITT V.42.

In order to identify the V-data block to assist the main controller circuit 313 to multiplex the packets for transmission at his level,
5 and to assist the remote site in recognizing and de-multiplexing the data packets, a V-data block is defined which includes a maximum of five V-data packets. The V-data block size and the maximum number of blocks are defined as follows:

- 10 The V-data block header = 80h;
- The V-data block size = 24;
- The maximum V-data block size = 5;

The V-data block has higher priority to be transmitted than C-data to ensure the integrity of the real-time voice transmission. Therefore, the main controller circuit 313 will check the V-data buffer first to determine
15 whether it will transmit V-data or C-data blocks. If V-data buffer has V-data of more than 69 bytes, a transmit block counter is set to 5 and the main controller circuit 313 starts to transmit V-data from the V-data buffer through the data pump circuit 311 onto the telephone line. Since the transmit block counter indicates 5 blocks of V-data will be transmitted in a continuous
20 stream, the transmission will stop either at finish the 115 bytes of V-data or if the V-data buffer is empty. If V-data buffer has V-data with number more than 24 bytes, the transmit block counter is set 1 and starts transmit V-data. This means that the main controller circuit will only transmit one block of V-data. If the V-data buffer has V-data with less than 24 bytes, the main
25 controller circuit services the transmission of C-data.

During the transmission of a C-data block, the V-data buffer condition is checked before transmitting the first C-data byte. If the V-data buffer contains more than one V-data packet, the current transmission of the C-data block will be terminated in order to handle the V-data.

30 Receive Mode

On the receiving end of the telephone line, the main controller circuit 313 operates at the multiplex control level to de-multiplex received

data to V-data and C-data. The type of block can be identified by checking the first byte of the incoming data blocks. Before receiving a block of V-data, the main controller circuit 313 will initialize a receive V-data byte counter, a backup pointer and a temporary V-data buffer pointer. The value
5 of the receiver V-data byte counter is 24, the value of the receive block counter is 0 and the backup pointer is set to the same value as the V-data receive buffer pointer. If the received byte is not equal to 80 hex (80h indicating a V-data packet), the receive operation will follow the current modem protocol since the data block must contain C-data. If the received
10 byte is equal to 80h, the main controller circuit 313 operating in receive mode will process the V-data.

For a V-data block received, when a byte of V-data is received, the byte of V-data is put into the V-data receive buffer, the temporary buffer pointer is increased by 1 and the receive V-data counter is decreased by 1. If
15 the V-data counter is down to zero, the value of the temporary V-data buffer pointer is copied into the backup pointer buffer. The value of the total V-data counter is added with 24 and the receive V-data counter is reset to 24. The value of the receive block counter is increased by 1. A flag to request service of V-data is then set. If the receive block counter has reached 5, the main
20 controller circuit 313 will not put the incoming V-data into the V-data receive buffer but throw it away. If the total V-data counter has reached its maximum value, the receiver will not put the incoming V-data into the V-data receive buffer but throw it away.

At the end of the block which is indicated by receipt of the
25 CRC check bytes, the main controller circuit 313 operating in the multiplex control level will not check the result of the CRC but instead will check the value of the receive V-data counter. If the value is zero, the check is finished, otherwise the value of the backup pointer is copied back into the current V-data buffer pointer. By this method, the receiver is insured to de-
30 multiplex the V-data from the receiving channel 24 bytes at a time. The main controller circuit 313 operating at the service level in the receive mode will monitor the flag of request service of V-data. If the flag is set, the main

controller circuit 313 will get the V-data from the V-data buffer and transmit it to the voice control DSP circuit 306 at a rate of 24 bytes at a time. After sending a block of V-data, it decreases 24 from the value in the total V-data counter.

5 Negotiation of Voice Compression Rate

The modem hardware component 20 incorporates a modified packet protocol for negotiation of the speech compression rate. A modified supervisory packet is formatted using the same open flag, address, CRC, and closing flag formatting bytes which are found in the CCITT V.42 standard data supervisory packet, as is well known in the industry and as is described in the CCITT Blue Book, volume VIII entitled Data Communication over the Telephone Network, 1989 referenced above. In the modified packet protocol embodiment, the set of CCITT standard header bytes (control words) has been extended to include nonstandard control words used to signal transmission of a nonstandard communication command. The use of a nonstandard control word does not conflict with other data communication terminals, for example, when communicating with a non-PCS (Personal Communications System) modem system, since the nonstandard packet will be ignored by a non-PCS system.

20 Table 22 offers one embodiment of the present invention showing a modified supervisory packet structure. Table 22 omits the CCITT standard formatting bytes: open flag, address, CRC, and closing flag; however, these bytes are described in the CCITT standard. The modified supervisory packet is distinguished from a V.42 standard packet by using a nonstandard control word, such as 80 hex, as the header. The nonstandard control word does not conflict with V.42 standard communications.

**TABLE 22: Modified
Supervisory Packet Structure**

30	80h	ID	LI	ACK	data	data	data
----	-----	----	----	-----	------	------	------

The modified supervisory packet is transmitted by the HDLC protocol in which data is transmitted in synchronous mode and checked by CRC error checking. The use of a modified supervisory packet eliminates the

need for an escape command sent over the telephone line to interrupt data communications, providing an independent channel for negotiation of the compression rate. The channel may also be used as an alternative means for programming standard communications parameters.

5 The modified supervisory packet is encoded with different function codes to provide an independent communications channel between hardware components. This provides a means for real time negotiation and programming of the voice compression rate during uninterrupted transmission of voice data and conventional data without the need for conventional escape
10 routines. The modified supervisory packet is encoded with a function code using several embodiments. For example, in one embodiment, the function code is embedded in the packet as one of the data words and is located in a predetermined position. In an alternate embodiment, the supervisory packet header signals a nonstandard supervisory packet and contains the compression
15 rate to be used between the sites. In such an embodiment, for example, a different nonreserved header is assigned to each function code. These embodiments are not limiting and other methods known to those skilled in the art may be employed to encode the function code into the modified supervisory packet.

20 Referring once again to Figure 1, a system consisting of PCS modem 20 and data terminal 10 are connected via phone line 30 to a second PCS system comprised of PCS modem 20A and data terminal 10A. Therefore, calling modem 20 initializes communication with receiving modem 20A. In one embodiment of the present invention, a speech compression
25 command is sent via a modified supervisory data packet as the request for speech compression algorithm and ratio negotiation. Encoded in the speech compression command is the particular speech compression algorithm and the speech compression ratio desired by the calling PCM modem 20. Several methods for encoding the speech compression algorithm and compression
30 ratio exist. For example, in embodiments where the function code is embedded in the header byte, the first data byte of the modified supervisory packet could be used to identify the speech compression algorithm using a

binary coding scheme (e.g., 00h for Vector Quantization, 01h for CELP+, 02h for VCELP, and 03h for TrueSpeech, etc.). A second data byte could be used to encode the speech compression ratio (e.g., 00h for 9.6Kbit/sec, 01h for 16Kbit/sec, 02h for 8Kbit/sec, etc.). This embodiment of the speech

5 compression command supervisory packet is shown in Table 23.

TABLE 23: Speech Compression Command Supervisory Packet

10	80h	ID	LI	ACK	Alghm	CRatio	data
----	-----	----	----	-----	-------	--------	------

Alternatively, as stated above, the function code could be stored in a predetermined position of one of the packet data bytes. Other function code encoding embodiments are possible without deviating from the scope and spirit of the present invention and the embodiments offered are not intended to be exclusive or limiting embodiments.

In either case, the receiving PCS modem 20A will recognize the speech compression command and will respond with an acknowledge packet using, for instance, a header byte such as hex 81. The acknowledge packet will alert the calling modem 20 that the speech compression algorithm and speech compression ratio selected are available by use of the ACK field of the supervisory packet shown in Table 23. Receipt of the acknowledge supervisory packet causes the calling modem 20 to transmit subsequent voice over data information according to the selected speech compression algorithm and compression ratio.

The frequency of which the speech compression command supervisory packet is transmitted will vary with the application. For moderate quality voice over data applications, the speech compression algorithm need only be negotiated at the initialization of the phone call. For applications requiring more fidelity, the speech compression command supervisory packet is renegotiated throughout the call to accommodate new parties to the communication which have different speech compression algorithm limitations or to actively tune the speech compression ratio as the quality of the communications link fluctuates.

Other embodiments provide a speech compression command supervisory packet encode varying transmission rates of the speech compression command supervisory packet and different methods of speech compression algorithm and compression ratio negotiation. Additionally, other
5 encoding embodiments to encode the supervisory packet speech compression algorithm and the speech compression ratio may be incorporated without deviating from the scope and spirit of the present invention, and the described embodiments are not exclusive or limiting.

A new supervisory packet may be allocated for use as a means
10 for negotiating multiplexing scheme for the various types of information sent over the communications link. For example, if voice over data mode is employed, there exist several methods for multiplexing the voice and digital data. The multiplexing scheme may be selected by using a modified supervisory packet, called a multiplex supervisory packet, to negotiate the
15 selection of multiplexing scheme.

Similarly, another supervisory packet could be designated for remote control of another hardware device. For example, to control the baud rate or data format of a remote modem, a remote control supervisory packet could be encoded with the necessary selection parameters needed to program
20 the remote device.

Those skilled in the art will readily appreciate that there exist numerous other unidirectional and bidirectional communication and control applications in which the supervisory packet may be used. The examples given are not limiting, but are specific embodiments of the present invention
25 offered for illustrative purposes.

The present inventions are to be limited only in accordance with the scope of the appended claims, since others skilled in the art may devise other embodiments still within the limits of the claims.

WE CLAIM:

1. A communication module for use with a personal computer, comprising:
 - 5 communications interface means connected for communicating to the personal computer for transferring data between the personal computer and the communications module;
 - telephone line interface means for connection to a telephone line;
 - voice interface means for receiving local voice signals from a local
 - 10 user and for conveying remote voice signals from a remote user to the local user;
 - full-duplex conversion means connected to the voice interface means for converting the local voice signals into outgoing digital voice data and for converting incoming digital voice data into the remote voice signals;
 - 15 digital signal processor means connected to the full-duplex conversion means for compressing the outgoing digital voice data into compressed outgoing digital voice data having one of a plurality of selectable compression rates and for decompressing compressed incoming digital voice data into the incoming digital voice data at one of said plurality of said selectable
 - 20 compression rates;
 - main control means connected for receiving the compressed outgoing digital voice data from the digital signal processor means, connected for receiving outgoing conventional digital data from the personal computer through the communications interface means, and operable for multiplexing
 - 25 the compressed outgoing digital voice data and the conventional digital data to produce multiplexed outgoing data; and
 - the main control means further operable for receiving multiplexed incoming data which contains incoming conventional digital data multiplexed with the compressed incoming digital voice data, for demultiplexing the
 - 30 incoming conventional digital data and the compressed incoming digital voice data, and for sending the incoming conventional digital data to the personal computer through the communications interface means and for sending the

compressed incoming digital voice data to the digital signal processor means;
and

the main control means further operable for negotiating the
compression rate at various times during the transmission of conventional
5 digital data multiplexed with the compressed digital voice data to change the
compression rate.

2. The module according to claim 1 wherein the digital signal processor
means is further operable for compressing the outgoing digital voice data into
10 compressed outgoing digital voice data by performing the steps of:

- a) removing any DC bias in the outgoing digital voice data to produce
a normalized outgoing digital voice signal;
- b) pre-emphasizing the normalized outgoing digital voice signal to
produce a pre-emphasized outgoing digital voice signal;
- 15 c) dividing the pre-emphasized outgoing digital voice signal into
segments to produce a current segment and a past segment;
- d) predicting the pitch of the current speech segment to form a pitch
prediction;
- e) calculating the gain of the pitch of the current speech segment to
20 form a prediction gain;
- f) reconstructing the past speech segment from a compressed past
segment to produce a reconstructed past segment;
- g) finding the innovation in the current speech segment by comparing
the pitch prediction to the reconstructed past segment to produce an error
25 signal;
- h) determining the maximum amplitude in the current speech segment;
- i) quantizing the error signal using one of a plurality of code books,
each code book corresponding to a different compression rate, the code books
being generated from a representative set of speakers and environments to
30 produce a minimum mean squared error matching the form of an index into
said one of said plurality of code book; and

j) recording the pitch prediction, the prediction gain, the maximum amplitude and the index into the selected code book in a packet as the compressed outgoing digital voice data.

- 5 3. The module according to claim 1 wherein the digital signal processor means is further operable for detecting silent periods in the outgoing digital voice data and for producing in response thereto a silence flag and wherein the main control means is further operable for transmitting outgoing conventional digital data on the telephone line when the silence flag indicates
- 10 the absence of voice information and wherein the main control means is further operable for multiplexing and transmitting both the compressed outgoing digital voice data and the outgoing conventional digital data on the telephone line when the silence flag indicates the presence of voice information.
- 15
4. A system for performing voice compression, comprising:
- voice interface means including a voice input device for receiving voice signals from a user;
- conversion means for converting the voice signals into digital voice
- 20 data;
- means for dividing the digital voice signal into segments and for serially producing therefrom a current voice segment and a past voice segment;
- means for determining the predicted gain of the current voice segment;
- 25 means for determining the pitch values of the current voice segment;
- coding means for predictive coding on the current voice segment by predicting the current voice segment as best as possible based on past recreated voice segments and for producing a difference signal value in response thereto;
- 30 means for selecting a compression rate for the speech compression;
- means including a plurality of code books stored in a memory for comparing the difference signal value to stored difference values stored one of

the plurality of code books stored in the memory and for locating the memory address of the closest match between the difference signal value and the stored difference values;

output means for providing a compression packet for each of the
5 segments, the packet including the memory address of the closest match between the difference signal value and the stored difference values, the predicted gain and the pitch values for each voice segment; and

output means further for providing a negotiation packet sent to determine the compression rate.

10

5. A method for compression speech information, comprising the steps of:

selecting a compression rate;

receiving a speech signal from a person, sampling and digitizing the
15 speech signal and dividing the digitized speech signal into a continuous stream of speech segments having a plurality of digital samples;

selecting a current speech segment from the continuous stream of speech segments and reconstructing a past speech segment from a previously compressed past segment to produce a reconstructed past segment;

20 comparing the current segment on a sample by sample basis to the reconstructed past segment to obtain innovation information and an indicator of the error in the comparing; and

storing the innovation information and the indicator of the error in the comparing in a data packet as a compressed data packet.

25

6. The method according to claim 5 wherein the step of comparing includes the following step:

determining the predicted gain of the current speech segment;

determining the pitch values of the current speech segment;

30 predictive coding the current speech segment by predicting the current speech segment as best as possible based on the recreated past speech segments and for producing a difference signal value in response thereto; and

comparing the difference signal value to stored difference values stored in a memory and for locating the memory address of the closest match between the difference signal value and the stored difference values.

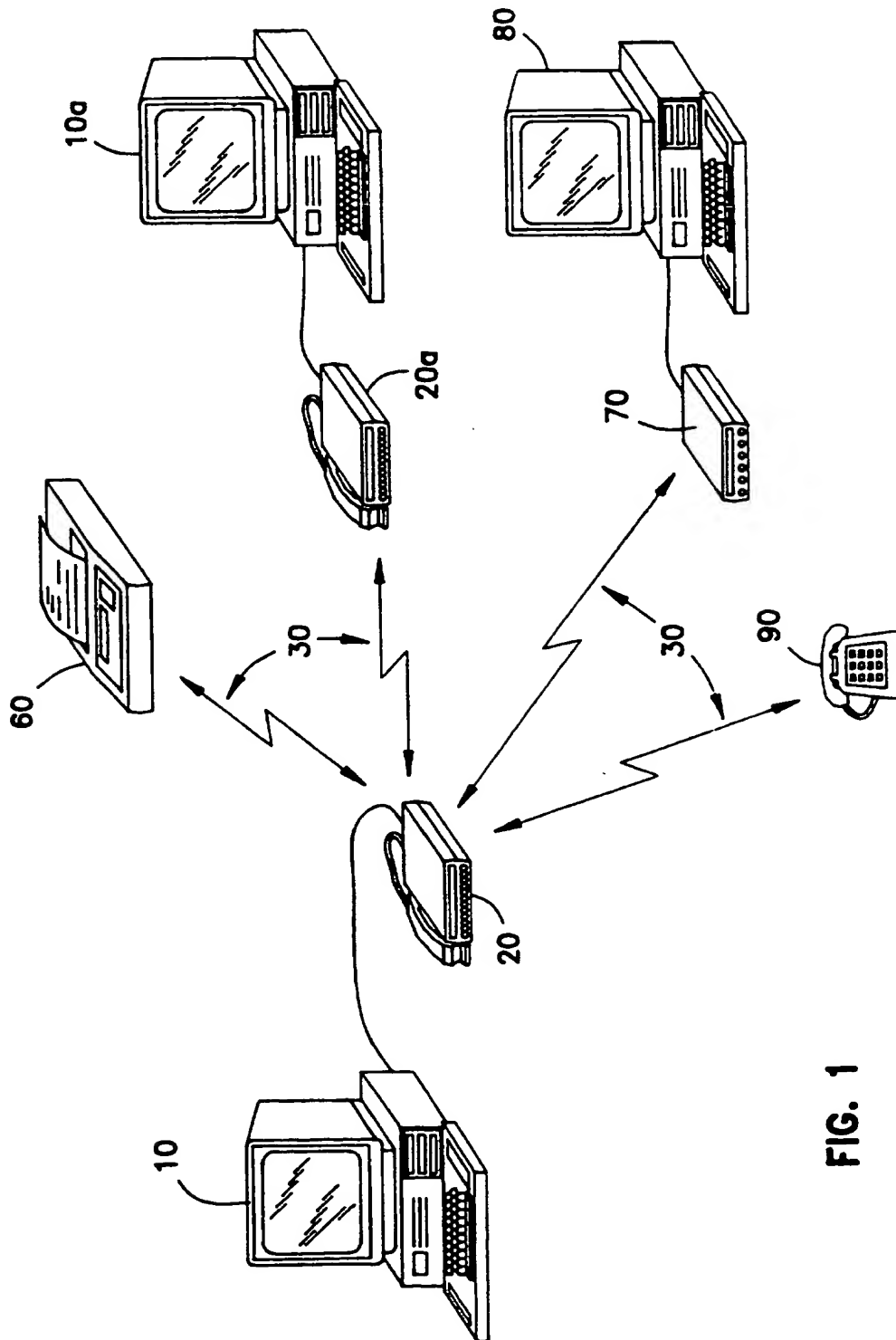


FIG. 1

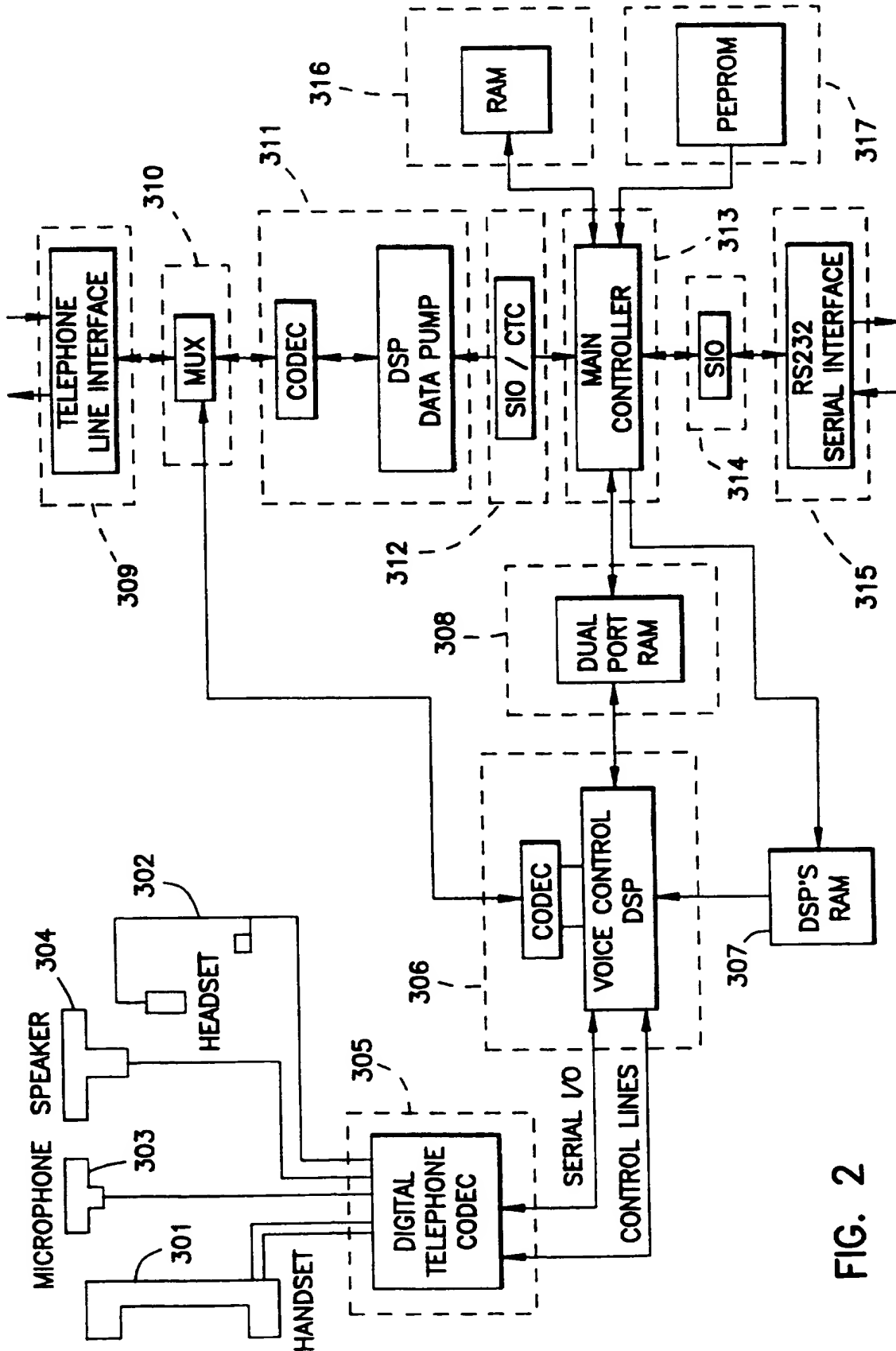


FIG. 2

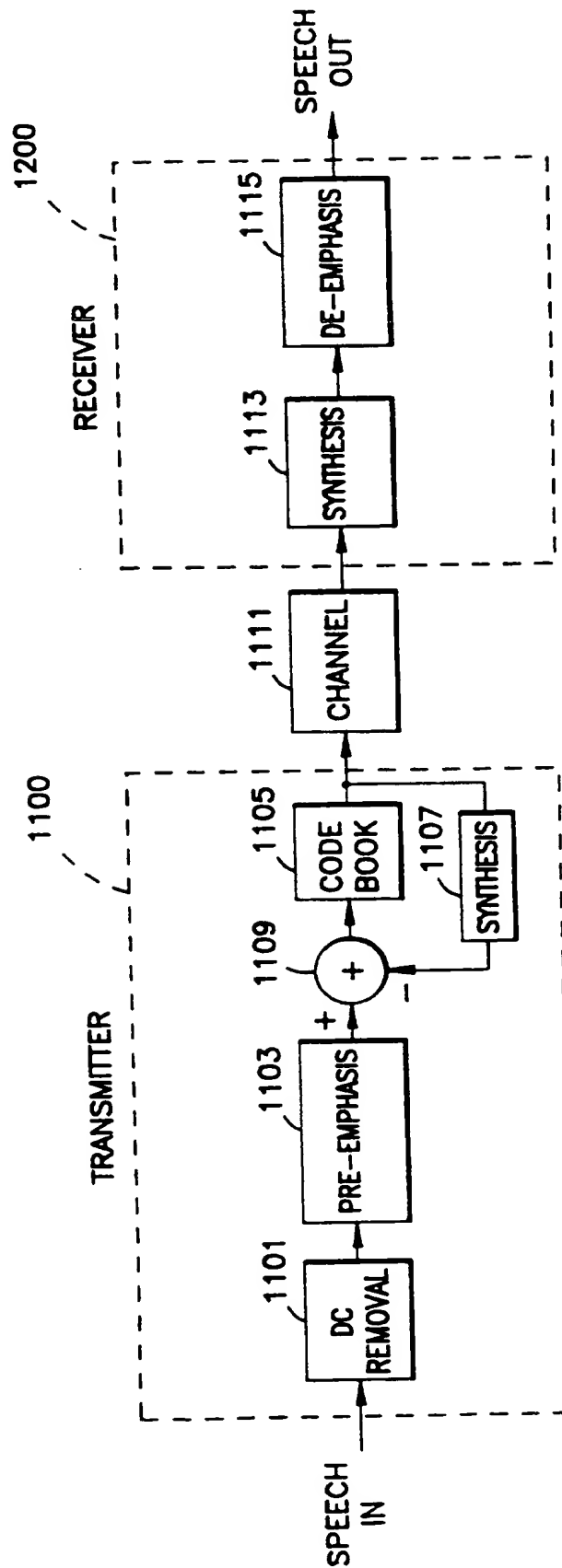


FIG. 3

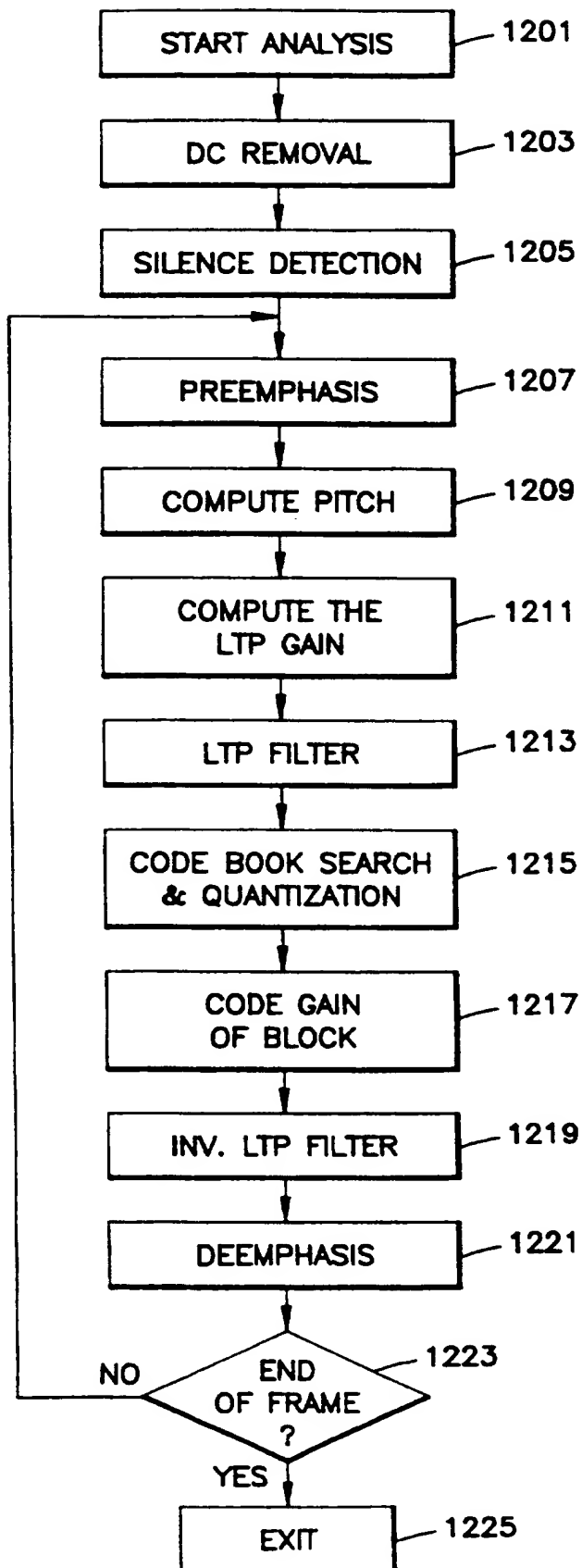


FIG. 4

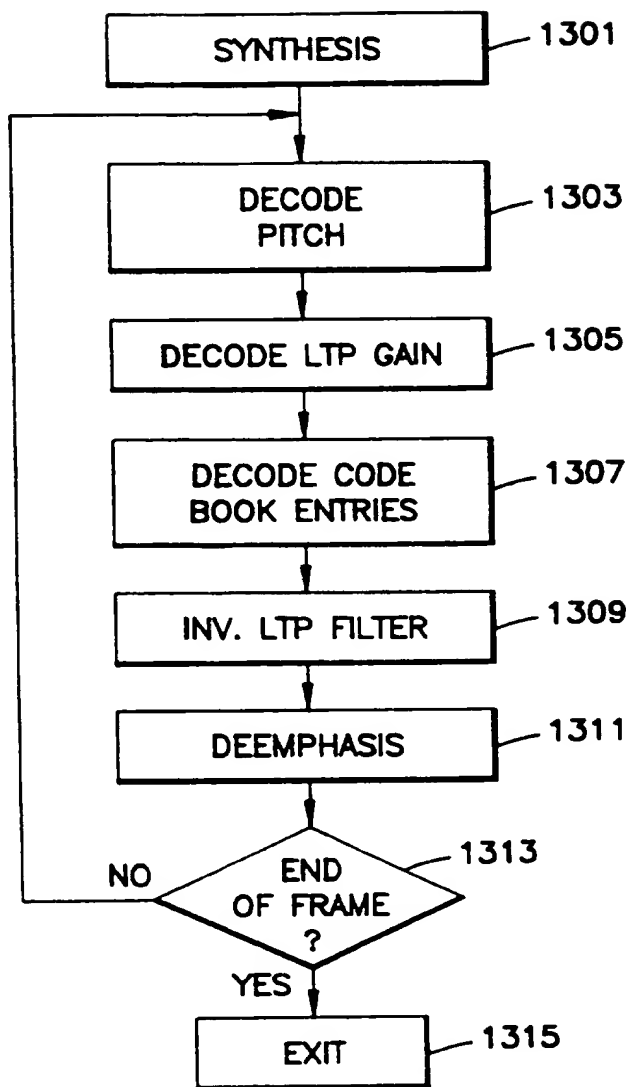


FIG. 5

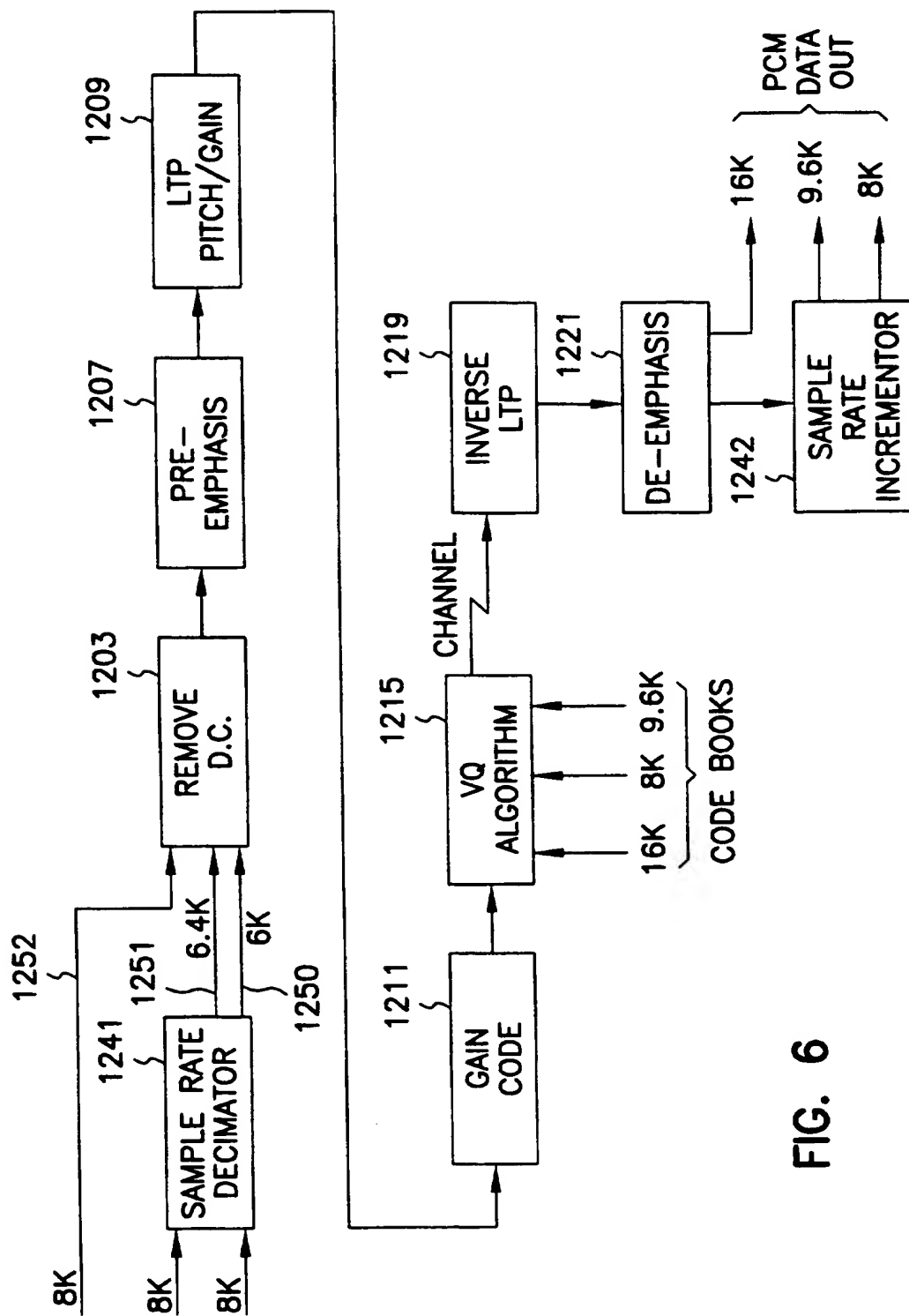


FIG. 6

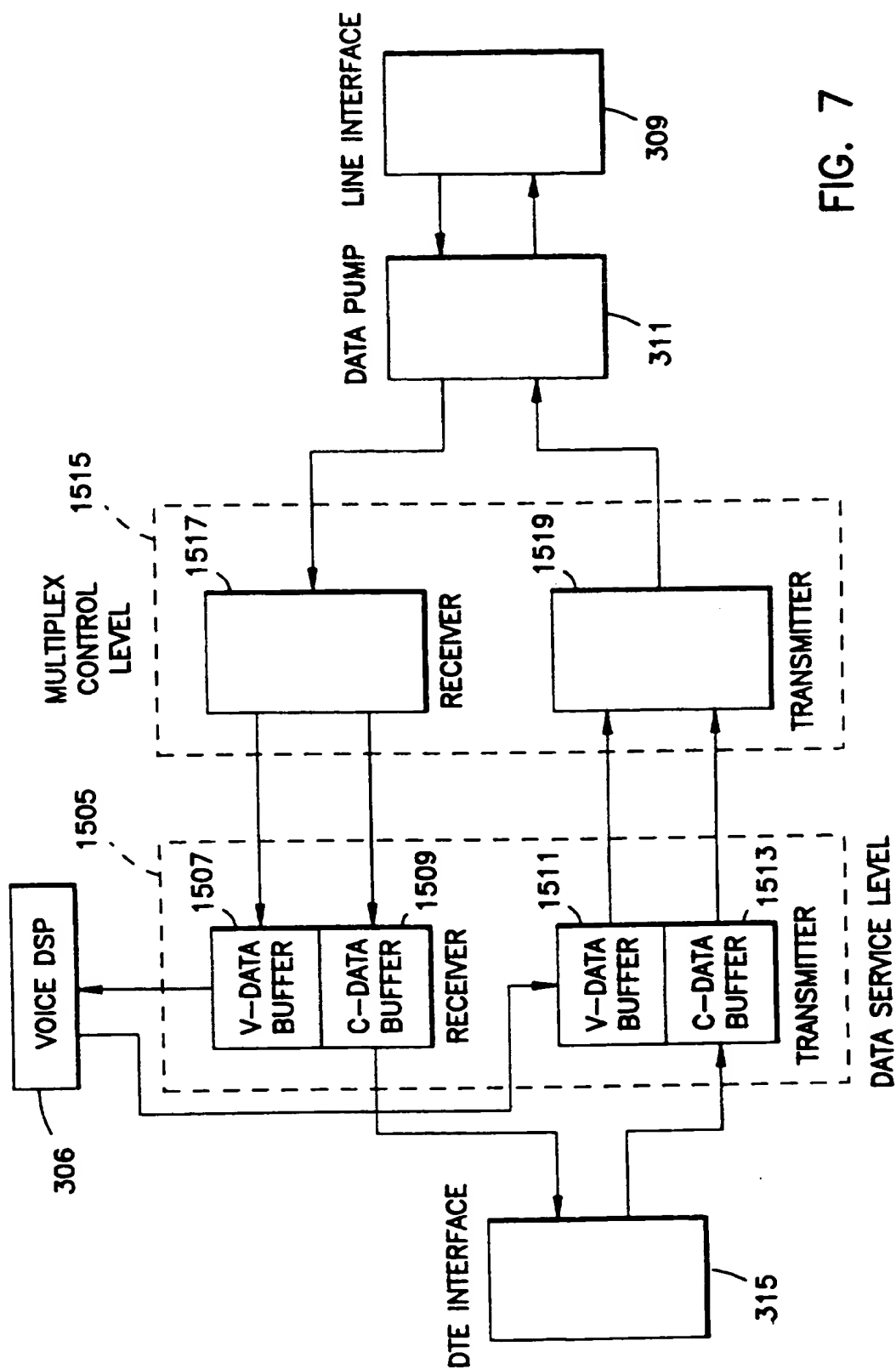


FIG. 7

INTERNATIONAL SEARCH REPORT

International Application No.

/US 95/14826

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04M11/06 G10L9/14

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 6 H04M G10L H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CA,A,2 104 701 (SHARMA ET AL.) 9 July 1994 cited in the application	1
Y	see page 53, line 21 - page 58, line 14	2,4,5
Y	see page 51, line 37 - page 52, line 25 ---	3
Y	EP,A,0 614 305 (HITACHI) 7 September 1994 see abstract see column 15, line 4 - line 46 ---	2,4,5
Y	US,A,5 121 385 (TOMINAGA ET AL.) 9 June 1992 see column 1 - column 2, line 27 ---	3
	-/-	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents:

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- *&* document member of the same patent family

Date of the actual completion of the international search

27 March 1996

Date of mailing of the international search report

10.04.96

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+ 31-70) 340-2040, Tx. 31 651 epo nl,
Fax (+ 31-70) 340-3016

Authorized officer

Lange, J

INTERNATIONAL SEARCH REPORT

International Application No
PCI/US 95/14826

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	IBM TECHNICAL DISCLOSURE BULLETIN, vol. 35, no. 6, November 1992, NEW YORK, US, pages 409-411, XP002000163 ANONYMOUS: "Method and Apparatus for the Statistical Multiplexing of Voice, Data, and Image Signals." ---	1
A	IBM TECHNICAL DISCLOSURE BULLETIN, vol. 27, no. 2, July 1984, NEW YORK, US, page 969 XP002000162 ANONYMOUS: "Speech Data Adaptive Multiplexer" ---	3
A	US,A,4 598 397 (NELSON ET AL.) 1 July 1986 see column 5, line 8 - line 63 -----	1-5

INTERNATIONAL SEARCH REPORT

International Application No.

SI/US 95/14826

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
CA-A-2104701	09-07-94	US-A- 5452289	19-09-95
		EP-A- 0630141	21-12-94
		US-A- 5453986	26-09-95
		US-A- 5471470	28-11-95
		US-A- 5500859	19-03-96

EP-A-0614305	07-09-94	JP-A- 6253072	09-09-94
		CN-A- 1100865	29-03-95
		US-A- 5359644	25-10-94

US-A-5121385	09-06-92	JP-A- 2079532	20-03-90
		JP-B- 7044511	15-05-95

US-A-4598397	01-07-86	AU-B- 571236	14-04-88
		AU-B- 2963384	19-11-84
		AU-B- 570983	31-03-88
		AU-B- 2965584	19-11-84
		CA-A- 1243787	25-10-88
		CA-A- 1242819	04-10-88
		EP-A- 0141852	22-05-85
		EP-A- 0142551	29-05-85
		WO-A- 8404436	08-11-84
		WO-A- 8404435	08-11-84
